RESEARCH **Open Access**

Feature selection followed by a novel residuals-based normalization that includes variance stabilization simplifies and improves single-cell gene expression analysis

Amartya Singh[1*] and Hossein Khiabanian[1,2,3]

*Correspondence:
as2197@scarletmail.rutgers.edu;
amartya.singh@rutgers.edu

[1] Center for Systems and Computational Biology, Rutgers Cancer Institute of New Jersey, Rutgers University, New Brunswick, NJ, USA
[2] Department of Pathology and Laboratory Medicine, Rutgers Robert Wood Johnson Medical School, Rutgers University, New Brunswick, NJ, USA
[3] Present Address: Regeneron Genetics Center, Regeneron Pharmaceuticals, Tarrytown, NY, USA

Abstract

Normalization is a crucial step in the analysis of single-cell RNA-sequencing (scRNA-seq) counts data. Its principal objectives are reduction of systematic biases primarily introduced through technical sources and transformation of counts to make them more amenable for the application of established statistical frameworks. In the standard workflows, normalization is followed by feature selection to identify highly variable genes (HVGs) that capture most of the biologically meaningful variation across the cells. Here, we make the case for a revised workflow by proposing a simple feature selection method and showing that we can perform feature selection before normalization by relying on observed counts. We highlight that the feature selection step can be used to not only select HVGs but to also identify stable genes. We further propose a novel variance stabilization transformation inclusive residuals-based normalization method that in fact relies on the stable genes to inform the reduction of systematic biases. We demonstrate significant improvements in downstream clustering analyses through the application of our proposed methods on biological truth-known as well as simulated counts datasets. We have implemented this novel workflow for analyzing high-throughput scRNA-seq data in an R package called Piccolo.

Introduction

Bulk RNA sequencing (RNA-seq) studies have led to a significant improvement in our understanding of gene expression profiles associated with healthy as well as diseased states of various tissue types. However, these studies only provide an averaged view at the tissue level in which subtle but crucial distinctions of the constituent cell-types and states are obscured. Rapid advances in single-cell RNA-seq (scRNA-seq) protocols and platforms over the past decade have now facilitated investigation of transcriptional profiles at the level of individual cells, thereby enabling identification of distinct cell-types and cell states [1–5], as well as stages of development and differentiation [6, 7].

In contrast to measurements on bulk tissues, single-cell measurements have significantly greater uncertainty due to the low amounts of starting material as well as low capture efficiencies of the protocols (typically, high-throughput protocols only capture between 5 and 20% of the molecules present in each cell [8]). As a result, even deeply sequenced datasets may have up to 50% zeros [9]. The high sparsity poses a significant challenge during the computational analysis. Early attempts to build statistical models to explain the relationship between the observed counts and the true underlying gene expression levels relied on zero-inflation models to explain the excess zeros. However, data generated using newer scRNA-seq protocols that rely on unique molecular identifiers (UMIs) have been shown to be sufficiently described using simpler statistical models that do not include zero inflation [10, 11]. In this paper, any reference to scRNA-seq data will specifically pertain to UMI counts data.

A critical step in the computational analyses of both bulk RNA-seq and scRNA-seq datasets is that of normalization. One of its key objectives is to reduce the biases introduced by technical sources or even biological sources such as cell cycle state, so that we can confidently identify true biological differences [9, 12–14]. Owing to the small amount of mRNAs captured per cell, the effect of these biases is more pronounced in the case of scRNA-seq data, further underscoring the need to reduce the impact of these biases on downstream analyses. Typically, normalization is performed by re-scaling the observed counts using cell-specific size factors to reduce the differences in sampling depths (total counts) between the cells. The scaled counts are then transformed with the help of a monotonic non-linear function (usually the logarithm function) to stabilize the variances of genes across different mean expression levels.

In the standard scRNA-seq workflow (implemented for instance in Seurat [15–18] and Scanpy [19]) normalization is followed by a feature selection step that focuses on identifying genes that capture most of the biological variation across the cells while eliminating genes that do not exhibit meaningful biological variation. This sequence of steps—normalization followed by feature selection—in the standard workflow appears quite reasonable, especially given the fact that differentially expressed genes can be identified reliably only after reducing the sampling depth differences between the cells. However, objective (i). identification of genes that are differentially expressed between groups of cells, is not the same as objective (ii). identification of highly variable genes (HVGs). While the identification of differentially expressed genes between distinct groups of cells requires that the sampling depth differences be reduced through normalization, we show that it isn't necessary to perform normalization in order to identify HVGs.

In this article, we propose a simple feature selection method that relies on a regression-based approach to estimate dispersion coefficients for the genes based on their observed counts. Using this method, we show that feature selection can be performed before normalization. Importantly, during the feature selection step we not only identify variable genes, but also shortlist stable genes. The variation in the counts of these stable genes is expected to primarily reflect the biases introduced by the technical sources, and can therefore be used to estimate cell-specific size factors in order to perform normalization. During normalization we also need to ensure variance stabilization, especially when relying on principal components analysis (PCA) for dimensionality reduction. Keeping this in mind, we propose a residuals-based normalization method that not only reduces the impact of sampling depth differences between the cells but simultaneously ensures variance stabilization by explicitly relying on a monotonic non-linear transformation (default choice is the *log* transformation).

We demonstrate significant improvements in downstream clustering analyses enabled by the application of our feature selection and normalization methods on biological truth-known as well as simulated counts datasets. Based on these results, we make the case for a revised scRNA-seq analysis workflow in which we first perform feature selection and subsequently perform normalization using our residuals-based approach. We have implemented this novel scRNA-seq workflow in an R package called `Piccolo`.

Results

Genes with small counts exhibit quasi-Poisson variance

Genes with low mean expression levels show Poisson-like variance of their counts

As pointed out by Sarkar and Stephens [11], a good starting point for understanding the nature of the scRNA-seq counts data is to recall that the observed counts reflect contributions from both the underlying expression levels of the genes as well as the measurement errors. This necessitates that the contributions from the two be carefully distinguished in order to better understand and explain the true biological variation. Based on their analysis, they found that a simple Poisson distribution sufficed to explain measurement error, while a simple Gamma distribution often sufficed to explain the variation in expression levels across cells. The observation model built using these two distributions yields the Gamma-Poisson (or negative binomial (NB)) distribution which is well-known as a plausible model to explain over-dispersed counts. Under this model, the mean-variance relationship is given by,

$$\sigma_{NB}^2 = \mu + \alpha_{NB}\mu^2 \tag{1}$$

where α_{NB} is the NB over-dispersion coefficient ($\alpha_{NB} = 0$ yields the familiar Poisson mean-variance relationship: $\sigma^2 = \mu$).

A more familiar expression for the NB mean-variance relationship is,

$$\sigma_{NB}^2 = \mu + \frac{\mu^2}{\theta} \tag{2}$$

where θ is referred to as the inverse over-dispersion coefficient.

Another mean-variance relationship closely related to the Poisson and the NB is the quasi-Poisson (QP) wherein,

$$\sigma_{QP}^2 = \alpha_{QP}\mu \tag{3}$$

where α_{QP} is the QP dispersion coefficient ($\alpha_{QP} = 1$ yields the familiar Poisson mean-variance relationship; $\alpha_{QP} > 1$ would be associated with counts over-dispersed with respect to the Poisson, while $\alpha_{QP} < 1$ would indicate counts under-dispersed with respect to the Poisson).

We began our investigation into the nature of UMI counts by considering the mean-variance relationship of counts for genes in a technical negative control data set [10] (hereafter referred to as `Svensson 1`). This data set consisted of droplets containing homogenous solutions of endogenous RNA as well as spike-in transcripts. The variability of counts in this data arises solely due to technical sources and is not attributable to any underlying biological source. The left panel in Fig. 1A shows the variance (σ^2) vs mean (μ) log-log plot for `Svensson 1`. Each dot in the plot corresponds to a gene. The colors of the dots reflect the local point density, with darker shades (deep blues) indicating low density and brighter shades (bright yellow) indicating high density. The black line depicts the variance expected under the Poisson model ($\sigma^2 = \mu$).

We begin by noting that the UMI counts are heteroskedastic in nature since the variances of the genes depend on the mean (larger variances corresponding to larger means). If we then move on to focus on genes with low mean expression levels (especially $\mu < 0.1$), we can see that the mean-variance relationship appears to be well-approximated by the Poisson model since their observed variances lie close to the black line. It's only for genes with higher mean expression levels (especially $\mu > 1$) that the variance of the observed counts exceed the variance expected under the Poisson model. For comparison with counts data with inherent biological variation, we looked at the NIH/3T3 fibroblast cell line data set [10] (hereafter referred to as `NIH/3T3`) and the 10X Genomics Peripheral Blood Mononuclear Cells (PBMC) 33k data set [20] (hereafter referred to as `PBMC 33k`). Even for these two datasets, most of the genes with low mean

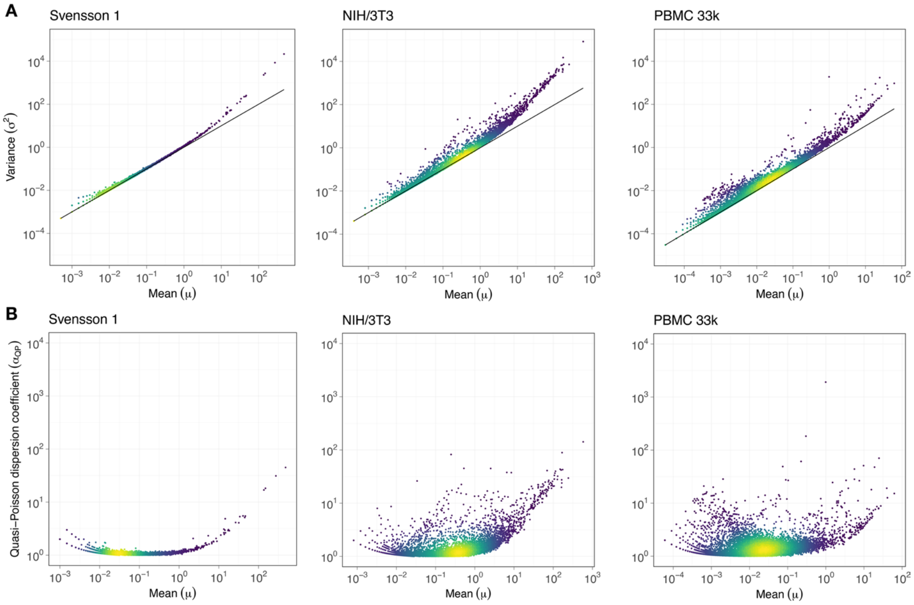

Fig. 1 Genes with low mean expression exhibit quasi-Poisson variance. In all the plots, each dot represents a gene and the color of the dots reflect the local point density, with brighter shades (yellow) indicating high density and darker shades (deep blue) indicating low density. **A** Variance (σ^2) vs mean (μ) log-log scatter plots for the Svensson 1 technical control (left panel), NIH/3T3 fibroblast cell line (center panel), and PBMC 33k (right panel) datasets. The solid black line corresponds to the Poisson model ($\sigma^2 = \mu$). For genes with low mean expression levels, the variance can be adequately described by the Poisson model. **B** Quasi-Poisson dispersion coefficients (α_{QP}) vs mean (μ) log-log scatter plots for the Svensson 1 (left panel), NIH/3T3 (center panel), and PBMC 33k (right panel) datasets. α_{QP} for each gene were estimated from the observed counts using a regression-based approach

expression levels exhibit Poisson-like mean-variance relationship (middle and right panels in Fig. 1A).

The Poisson-like nature of the mean-variance relationship at low expression levels can be made clearer with the help of a simple example. Consider a gene with $\mu = 0.01$. Let us assume that the counts of this gene are actually NB distributed and that $\alpha_{NB} = \frac{1}{\theta} = 1$ is the true estimate of the over-dispersion coefficient. The variance of the counts will then be $\sigma^2_{NB} = 0.01 + 1 * (0.01)^2 = 0.0101$. In comparison, the expected variance under the Poisson model will be $\sigma^2_{Pois} = 0.01$. The percent difference between the two variances is a mere 1%. Thus, even if we approximated the variance for the counts of this gene with that predicted by the simple Poisson model, the estimated variance would differ by only 1% from the true NB variance. The percentage difference between the Poisson and the NB variances would of course increase for larger values of α_{NB}, however, estimates of α_{NB} for real biological datasets typically range between 0.01 and 1 [21, 22].

We can formalize the discussion above with the help of Eqs. 1 and 3. We note that when $\alpha_{NB}\mu << 1$,

$$\sigma^2_{NB} = (1 + \alpha_{NB}\mu)\mu = \sigma^2_{QP} = \alpha_{QP}\mu \approx \sigma^2_{Pois} \tag{4}$$

where we defined α_{QP} in terms of α_{NB} and μ as $\alpha_{QP} = 1 + \alpha_{NB}\mu$ (see Appendix for a discussion on viewing QP variance as a special case of NB variance).

Thus, we conclude that counts of genes with low mean expression levels and moderate over-dispersion (such that $\alpha_{NB}\mu << 1$) exhibit variances that do not deviate significantly from the variances predicted under the Poisson model.

QP dispersion coefficients can be obtained using a regression-based approach

For the NB distribution, the usual approach is to use maximum-likelihood estimation (MLE) to obtain estimates for the over-dispersion parameter (α_{NB}). This approach apart from being computationally intensive has the weakness that if in fact the distribution is not NB, the maximum-likelihood estimator is inconsistent. Cameron and Trivedi proposed a regression-based test that offers a more robust alternative by requiring only the estimates of the mean and variance for each gene [23, 24]. The test is set up to estimate over-dispersion beyond the null model (Poisson distribution) by specifying the alternate model in the form of a scalar multiple of a function of the mean,

$$Var[x] = \alpha E[x] \tag{5}$$

where the scalar multiple (α) is estimated using least squares regression (see Appendix).

The QP mean-variance relation (see Eq. 3) corresponds precisely to such an alternative hypothesis in which the variance is simply a scalar multiple (α_{QP}) of the mean. This enables us to use this simple yet robust regression-based approach to estimate the QP dispersion coefficients (α_{QP}) for each gene by simply relying on the estimates of their mean and variance.

QP variance for counts of genes with low mean expression is simply due to the observed counts being small and is not biological in origin

We obtained estimates for α_{QP} for all genes using the regression-based approach. Based on these estimates, we filtered out genes that were under-dispersed compared to the Poisson model ($\alpha_{QP} < 1$). For the remaining genes, we plotted α_{QP} vs μ log-log plots (Fig. 1B). For `Svensson 1` (left panel in Fig. 1B), it is evident that for genes with low mean expression levels their α_{QP} values lie close to 1 (10^0). In particular, we note that the mean of the α_{QP} for genes with $\mu < 0.1$ is 1.0474. This shows that for these genes with low mean expression their α_{QP} values are close to 1 which is consistent with the observation that the counts of these genes exhibit Poisson-like variance. Since `Svensson 1` is a technical control data set, this observation further supports the simple Poisson distribution as an appropriate model for explaining measurement error, particularly for genes with low mean expression levels. Furthermore, genes with $\mu < 0.1$ do not appear to exhibit any dependence on μ. We evaluate this quantitatively by using the non-parametric Kendall's rank and Spearman's rank correlation tests to determine whether there is a statistical dependence between α_{QP} and μ values for genes with $\mu < 0.1$ (see Methods). Both tests evaluate how well the relationship between two variables can be described using a monotonic function. For both tests, the correlation coefficients - τ and ρ respectively—indicate a statistical dependence if the values are close to $+1$ or -1, while values of τ or ρ closer to 0 indicate the absence of such a statistical dependence. The resultant correlation coefficient values of $\tau = 0.04611706$ $(p = 0.001244)$ and $\rho = 0.0880$ $(p = 3.475E - 05)$, support the assertion that there is no statistical dependence between α_{QP} and μ values for genes with low mean expression levels.

For `NIH/3T3` (middle panel in Fig. 1B) and `PBMC 33k` (right panel in Fig. 1B), despite greater variability in α_{QP} due to the inherent biological variability in the data, similar observations were made—namely that the α_{QP} values are close to 1 for genes with $\mu < 0.1$ and that there was a lack of dependence between α_{QP} and μ for those genes. For `NIH/3T3`, the mean of the α_{QP} for genes with $\mu < 0.1$ is 1.1388, and both Kendall's correlation coefficient $\tau = 0.0644$ $(p = 1.28E - 05)$ and Spearman's correlation coefficient $\rho = 0.1014$ $(p = 4.084e - 06)$ indicate that there is no statistical dependence between α_{QP} and μ for genes with $\mu < 0.1$. For `PBMC 33k`, the mean of the α_{QP} for genes with $\mu < 0.1$ is 1.3710, and the Kendall's correlation coefficient $\tau = 0.1349$ $(p < 2.2E - 16)$ and Spearman's correlation coefficient $\rho = 0.2035$ $(p < 2.2E - 16)$ once again suggesting that there is no significant statistical dependence between the α_{QP} values and μ for genes with low mean expression levels.

To summarize, we observed that the counts for genes with low mean expression levels exhibit QP variance (see Appendix for a discussion on how the lack of dependence between α_{QP} and μ for genes with low mean expression levels manifests as a non-decreasing relationship between their θ and μ). It is important to highlight here that we observed this relationship not just in the biological datasets (`NIH/3T3` and `PBMC 33k`) but also in the technical control data set (`Svensson 1`), suggesting that this relationship is not biological in origin and can be understood more simply in terms of the fact that for genes with small counts the variance of those counts can barely exceed the variance expected under the Poisson distribution.

Feature selection can be performed before normalization

Standard scRNA-seq workflow is based on an assumption that reflects a confusion between the distinct objectives of identification of variable genes and differentially expressed genes

In the standard scRNA-seq workflow, identification of HVGs—genes that exhibit greater variability in counts compared to other features in the data set—is performed only after normalization. This particular sequence in the workflow is based on the assumption that unless the systematic biases are reduced or eliminated, we cannot reliably identify features that best capture the biological variability inherent in the data. A more careful examination reveals that this assumption reflects a confusion between two fundamentally distinct objectives. The first objective— identification of HVGs - as mentioned above, is about identifying genes that exhibit higher variability of counts compared to other features in the data set; we expect that these genes capture most of the biological variability in the data set. The second objective—identification of differentially expressed (DE) genes—is about identifying genes that exhibit differences in their expression levels between distinct sets of cells. For identifying DE genes, it is indeed imperative that the systematic biases are reduced or eliminated in order to identify genes that truly reflect actual biological differences between the distinct groups of cells being compared. However, it is possible to identify HVGs without first accounting for the systematic biases through normalization since these biases owing to their systematic nature are expected to manifest as additional but *consistent* sources of variation for the counts of genes across cells. This is in fact the assumption underlying the normalization approaches that rely on estimation of cell-specific size factors to re-scale and adjust the observed counts.

We can state the expectation discussed above in terms of the QP dispersion coefficients. Suppose we have gene A and gene B with comparable mean expression levels ($\mu_A \approx \mu_B$) such that,

$$\tilde{\alpha}_{QP}^{A} > \tilde{\alpha}_{QP}^{B}$$

where $\tilde{\alpha}_{QP}^{A}$ and $\tilde{\alpha}_{QP}^{B}$ are the QP dispersion coefficients of gene A and gene B in the hypothetical case where there is no systematic bias. Our expectation is that even in the presence of systematic biases, the relative magnitudes of the dispersion coefficients for the bias-affected counts will be such that,

$$\alpha_{QP}^{A} > \alpha_{QP}^{B}$$

where α_{QP}^{A} and α_{QP}^{B} are the respective QP dispersion coefficients obtained from the observed counts of genes A and B. Thus, the expectation is that genes that exhibit high variability in their counts due to underlying biological differences will exhibit high variability even in the presence of systematic biases. We can test whether this is a reasonable expectation by introducing systematic biases into a given data set and then verifying whether the genes we identify as variable for the modified data set are consistent with the ones identified for the original data set. Before proceeding to perform such a test, however, we first need to lay down a method to identify HVGs.

Feature selection method based on QP dispersion coefficients identifies HVGs and stable genes

As discussed earlier, distinguishing between measurement and biological processes by separately modeling their contributions to observed counts enables a better understanding and interpretation of the observed counts [11]. It also provides a simple and straightforward basis for identifying and filtering out features that are relatively uninformative prior to any downstream analysis: if the measurement model sufficiently describes the observed counts for a given gene, then no meaningful biological inference can be drawn based on the counts of that gene. This is so since all the variation in the counts for that gene can be attributed to the measurement process itself, with negligible or no contribution from any biological process. We call such genes whose counts are adequately described by the measurement model as *biologically uninformative*. Since a simple Poisson distribution suffices to explain measurement error, using the estimates for α_{QP} we can easily identify genes that are likely to be biologically uninformative—genes with $\alpha_{QP} \leq 1$ do not exhibit over-dispersion with respect to the Poisson measurement model and can be filtered out.

While we can simply rely on the α_{QP} to identify biologically uninformative genes, identification of HVGs based on just the magnitudes of α_{QP} would result in a bias towards genes with higher mean expression levels (see Fig. 1B). In order to address this and ensure that there is no preferential selection of genes with higher mean expression levels, we propose the following approach to shortlist HVGs:

- Group the genes into bins (default is 1000 bins) based on their mean expression levels; each bin contains approximately the same number of genes with comparable mean expression levels
- Sort the genes within each bin into quantiles based on their α_{QP}
- Obtain the α_{QP} corresponding to the reference quantile within each bin (default reference is the 10th quantile)—we refer to this as $\alpha_{QP(Reference|Bin)}$
- Calculate $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ for each gene - larger values indicate greater over-dispersion

We illustrate the binning process involved in our feature selection method for `Svensson 1` (Fig. 2A). For ease of illustration we show 10 bins (see Appendix and Additional file 1: Fig. S13 for a discussion on how the number of bins impact the identification of HVGs), where each bin corresponds to a segment or region between the vertical dashed lines. Since `Svensson 1` is a technical control data set, we don't expect to see much variability in the expression levels of the genes, and indeed in Fig. 2C we can see that there are just two genes (spike-in transcripts ERCC-00074 and ERCC-00130) that exceed the threshold of 20 for $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ (red horizontal dashed line). For `PBMC 33k`, in contrast, we observe 15 genes that exceed the threshold of 20 for $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ (Fig. 2D). The threshold of 20 was picked simply to illustrate how the HVGs may be shortlisted. In practice, we can rank the genes based on the magnitudes of $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ and shortlist the top 3000 genes. An important point to highlight here is that the shortlisted HVGs do not exclude genes with low mean expression—note the inset panel in the top left corner of Fig. 2D, where we show the histogram

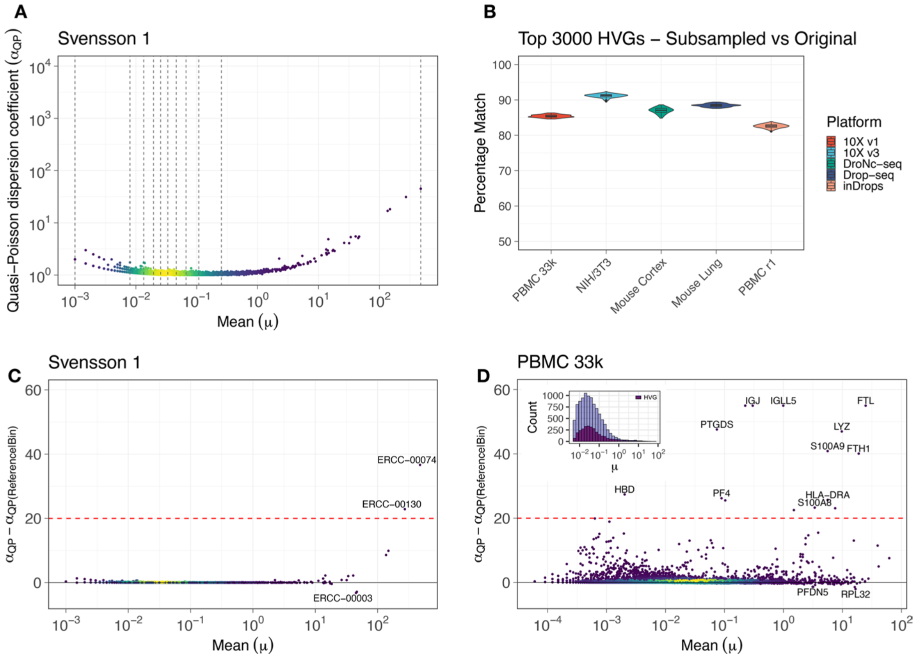

Fig. 2 Feature selection can be performed before normalization based on dispersion coefficients estimated using the observed counts. **A** Quasi-Poisson dispersion coefficient (α_{QP}) vs mean (μ) log-log scatter plots for genes that exhibited over-dispersion with respect to the Poisson ($\sigma^2 > \mu$) for the Svensson 1 technical control data set. Each dot represents a gene. The dashed gray vertical lines illustrate how the genes are binned based on their mean expression levels (in this figure, there are 10 bins). Genes within adjacent pairs of dashed vertical lines belong to the same bin. For each bin, the default choice of $\alpha_{QP(Reference|Bin)}$ is the α_{QP} corresponding to 10th quantile within the bin. **B** Violin-box plots showing the percentage match between the top 3000 highly variable genes (HVGs) identified by our feature selection method for the original (unsubsampled) and corresponding 100 subsampled datasets. The original 5 datasets (PBMC 33k, NIH/3T3 cell line, Mouse cortex, Mouse lung, and PBMC r1) were obtained using different single-cell platforms—10X Chromium v1, 10X Chromium v3, DroNC-seq, Drop-seq, and inDrops respectively. The box/violin plots are colored according to the platforms. For all 5 datasets, more than 80% of the top 3000 HVGs shortlisted for the subsampled datasets matched the top 3000 HVGs of the original datasets. **C** $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ vs mean (μ) linear-log scatter plot for genes that exhibited over-dispersion with respect to the Poisson ($\sigma^2 > \mu$) for the Svensson 1 data set. The dashed red horizontal line illustrates the threshold to shortlist HVGs. Genes with $\alpha_{QP} - \alpha_{QP(Reference|Bin)} > 20$ in this case are shortlisted as HVGs. On the other hand, genes with $\alpha_{QP} - \alpha_{QP(Reference|Bin)} < 0$ are shortlisted as stable genes. **D** $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ vs mean (μ) linear-log scatter plot for genes that exhibited over-dispersion with respect to the Poisson ($\sigma^2 > \mu$) for the PBMC 33k data set. Labeled genes with $\alpha_{QP} - \alpha_{QP(Reference|Bin)} > 20$ are shortlisted as HVGs. On the other hand, genes with $\alpha_{QP} - \alpha_{QP(Reference|Bin)} < 0$ are shortlisted as stable genes (some labeled in figure). Inset in top left corner—light colored histogram corresponding to all genes with $\alpha_{QP} > 1$, and dark colored histogram corresponding to the top 3000 HVGs. Note that the HVGs are shortlisted across the different expression levels with no preferential selection based on the mean expression level of the genes

based on the mean expression levels for the HVGs (darker shade) together with the histogram for all genes with $\alpha_{QP} > 1$ (lighter shade).

Aside from identifying variable genes, we can also shortlist genes that do not exhibit much variability in their counts. We refer to these genes as *stable genes*. Identification of such genes is very useful since we expect that the variability in the counts of these genes is primarily attributable to the measurement process and is not confounded by biological differences between the cells. Thus, we can rely on these genes to obtain more reliable estimates for the cell-specific size factors in order to reduce the impact of the differences in sampling depths. Based on our feature selection method, genes with $\alpha_{QP} - \alpha_{QP(Reference|Bin)} < 0$ are shortlisted as stable genes. For `PBMC 33k`, two of these genes (*PFDN2* and *RPL32*) are labeled in Fig. 2D below the horizontal black line at $\alpha_{QP} - \alpha_{QP(Reference|Bin)} = 0$.

HVGs can be consistently identified despite the introduction of systematic biases

Having introduced the feature selection method, we can now test the expectation that the HVGs can be identified despite the presence of systematic biases. To perform the test, we picked 5 UMI counts datasets obtained from different platforms: `PBMC 33k` (10X Genomics Chromium v1), `NIH/3T3` (10X Genomics Chromium v3), `Mouse Cortex` r1 (DroNC-seq) [8], `Mouse Lung` (Drop-seq) [25], and `PBMC r1` (inDrops) [8]; r1 denotes replicate 1. For each of these datasets, we randomly picked 30% of the cells and subsampled the counts in those cells to a fraction of the original total count. The fractions were allowed to take one of the following values - 0.3, 0.4, 0.5, 0.6, 0.7 - and were picked randomly for each cell. We did this 100 times for each of the 5 datasets. Using our feature selection method, we shortlisted the top 3000 HVGs in each of the subsampled datasets and compared with the top 3000 HVGs in the respective original (unsubsampled) datasets to see how many of the top 3000 HVGs matched between the two. We observed that on average more than 80% of the top 3000 HVGs shortlisted for the subsampled datasets matched the top 3000 HVGs obtained from the original datasets across the different platforms and tissue types (Fig. 2B). Thus, despite the introduction of random systematic biases there is good agreement between the HVGs obtained for the original datasets and the HVGs obtained for corresponding datasets with the introduced biases. This supports our assertion that feature selection can be performed prior to normalization.

Normalization can be performed using a residuals-based approach that includes variance stabilization

Cell-specific size factors should be estimated using stable genes

The basic objective of normalization is to reduce systematic biases introduced due to technical or potentially uninteresting biological sources (such as cell size, cell cycle state) before further downstream analyses such as clustering and differential expression. The most common approach to reduce such systematic biases is to scale the counts within each cell by cell-specific *size factors* [13, 26]. The simplest estimates are given by,

$$SF_c = \frac{\sum_g X_{gc}}{(\sum_c \sum_g X_{gc})/C} \tag{6}$$

where X_{gc} is the observed count of gene g in cell c, and C is the total number of cells. The numerator in Eq. (6) is the total UMI count in cell c (N_c), while the denominator is the mean of the total counts of the cells.

There is an intimate link between the estimates of size factors given by Eq. (6) and the estimates for expected means ($\hat{\mu}_{gc}$) under the assumption that the counts are Poisson distributed. The size factors as given by Eq. (6) should be viewed as estimates under the approximation that the counts are Poisson distributed (see Appendix). Keeping this in mind, we argue that it is most appropriate to calculate the size factors by relying on the stable genes identified using our feature selection method. The counts of these genes do not exhibit significant over-dispersion compared to the Poisson, and as discussed earlier, we expect that the primary source of the variability of their counts is the measurement process which is what we are trying to account for with the help of the size factors. Therefore, we propose the following refinement to the estimation of the size factors,

$$SF_c = \frac{\sum_{g \in Stable\,g} X_{gc}}{\left(\sum_c \sum_{g \in Stable\,g} X_{gc}\right)/C} \tag{7}$$

where $Stable\,g$ refers to the set of stable genes.

Limitations of widely used normalization methods

In the standard scRNA-seq workflow, normalization and feature selection are typically followed by dimensionality reduction using principal components analysis (PCA). An important step prior to PCA is to ensure variance stabilization by transforming the size factor adjusted counts using the *acosh*, *log*, *sqrt* functions etc. This ensures that genes with higher expression levels (and as a consequence larger variance) do not contribute disproportionately to the overall variance as evaluated through PCA. The *log*-based variance stabilization transformation (hereafter referred to as `logSF`) is the most popular method according to which the transformed counts are given by,

$$X'_{gc} = log\left(\frac{X_{gc}}{SF_c} + 1\right)$$

where the pseudocount of 1 ensures that the *log* transformation works with zero counts, and in fact returns zeros for these counts even after transformation. However, the `logSF` normalization is not very effective since the total counts of the cells (N_c) shows up as a primary source of variation in PCA even after normalization (see Additional file 1: Fig. S15). This can be traced to the fact that under this transformation, the zeros remain zeros while only the non-zero counts are scaled according to the size factors. Systematic differences in the number of zero counts between the cells can therefore be identified as a major source of variation even after transformation [27, 28].

Residuals-based approaches proposed by Hafemeister et al. [20], Townes et al. [27], and more recently by Lause et al. [21] provide alternatives that lead to much more effective normalization (see Additional file 1: Fig. S16). The Pearson residuals under the NB model are approximately given by,

$$r_{gc} = \frac{X_{gc} - \hat{\mu}_{gc}}{\sqrt{\hat{\mu}_{gc} + \hat{\alpha}_g \hat{\mu}_{gc}^2}} \tag{8}$$

where X_{gc} is the observed count of gene g in cell c, $\hat{\mu}_{gc}$ is the estimated mean of gene g in cell c, and $\hat{\alpha}_g$ is the estimated over-dispersion parameter for gene g.

The increased effectiveness of normalization with the Pearson residuals can be attributed to $\hat{\mu}_{gc}$ taking into account the systematic differences in the total counts between the cells. Unlike the case with `logSF` normalization where zero counts are transformed back to zeros, the zero counts are instead transformed to negative residual values whose magnitudes vary depending on the total counts of the respective cells.

The rationale underlying the residuals-based approach is that the null model should correspond to the measurement process so that the residuals provide estimates for deviations away from the expectations under the measurement model. However, the sparsity and the skewed nature of the counts distributions pose significant challenges to achieving effective variance stabilization with the help of residuals-based methods. The lack of variance stabilization becomes especially noticeable for genes that are robustly expressed in only a subset of cells while showing negligible expression in the rest of the cells (such genes would be considered as *markers* of the specific cell sub-populations in which they are expressed) [22] (see Appendix for more discussion on this). Furthermore, for genes with very low mean expression levels (and as a consequence extremely small estimated standard deviations) even cells with just 1 or 2 UMI counts sometimes end up with unusually large residual values that are then addressed through heuristic approaches [20, 21, 29].

Novel residuals-based normalization that includes variance stabilization

Keeping in mind the limitations of the widely used normalization methods discussed above, we propose a conceptually simple residuals-based normalization method that reduces the influence of systematic biases by relying on size factors estimated using stable genes while simultaneously ensuring variance stabilization by explicitly relying on a variance stabilization transformation.

In order to motivate our approach, we begin by pointing out that z-scores are simply Pearson residuals corresponding to the normal distribution. Assuming counts, Y_{gc}, that are normally distributed,

$$Y_{gc} \sim N(\mu_g, \sigma_g^2)$$

the MLEs for μ_g and σ_g^2 are given by, $\hat{\mu_g} = (\sum_c Y_{gc})/C$ and $\hat{\sigma}_g^2 = (\sum_c (Y_{gc} - \hat{\mu_g})^2)/C$, respectively. The corresponding Pearson residuals-based on these estimates are given by,

$$r_{gc} = \frac{Y_{gc} - \hat{\mu}_g}{\hat{\sigma_g}} = z_{gc} \tag{9}$$

which as already noted above correspond to z-scores (for simplicity, we assumed that there are no differences in total counts between the cells).

In order to compute z-scores for our data, we first need to apply a variance stabilization transformation to the raw counts (X_{gc}) to bring their distribution closer to the normal distribution. The variance stabilization transformation can be performed using monotonic non-linear functions, $g(X)$, such that the transformed counts are given by,

$$Y = g(X)$$

To compute the residuals, we need estimates for the means and variances of Y based on estimates for means and variances of X. We can arrive at approximations for both using a Taylor series expansion around $X = \mu$ (see Appendix). In particular for $g(X) = log(X + 1)$, the first order approximations of the mean and variance are given by (the first-order approximation to the variance of a transformed random variable is also known as the Delta method attributed to R. A. Dorfman [30]),

$$E[log(X+1)] \approx log(\mu+1) \tag{10}$$

$$Var[log(X+1)] \approx \frac{1}{(\mu+1)^2}\sigma^2 \tag{11}$$

At this point we need estimates for the means ($\hat{\mu}_{gc}$) and the variances ($\hat{\sigma}^2_{gc}$) that account for the systematic biases. Note, that the usual estimate for the mean expression level of gene g based on the observed counts ($\mu_g = \frac{\sum_c X_{gc}}{C}$) includes biological as well as technical effects. However, we are interested in inferring the mean expression level and variance that is primarily reflective of the underlying biology. This can be accomplished by relying on the size factors (Eq. (7)) to adjust the observed counts of the respective cells and then obtaining the estimates for mean and variance based on the scaled counts. Thus,

$$\begin{aligned}\tilde{\mu}_g =& \frac{1}{C}\sum_c \frac{X_{gc}}{SF_c}\\ \tilde{\sigma}^2_g =& \frac{1}{C-1}\sum_c (\frac{X_{gc}}{SF_c} - \tilde{\mu}_g)^2\end{aligned}$$

Using the estimates for mean and variance for gene g, we get the following estimates for mean and variance of gene g in each cell c,

$$\hat{\mu}_{gc} = SF_c\tilde{\mu}_g \tag{12}$$

$$\hat{\sigma}^2_{gc} = SF^2_c\tilde{\sigma}^2_g \tag{13}$$

With these estimated means ($\hat{\mu}_{gc}$) and variances ($\hat{\sigma}^2_{gc}$) that account for the systematic biases, from Eqs. (10) and (11) we get,

$$E[log(X_{gc}+1)] \approx log(\hat{\mu}_{gc}+1) \tag{14}$$

and

$$Var[log(X_{gc}+1)] \approx \frac{1}{(\hat{\mu}_{gc}+1)^2}\hat{\sigma}^2_{gc} \tag{15}$$

Based on these first-order approximations for means and variances under the $log(X+1)$ transformation, we now define our z-scores based normalization,

$$Z'_{gc} = \frac{log(X_{gc}+1) - log(\hat{\mu}_{gc}+1)}{\hat{\sigma}_{gc}/(\hat{\mu}_{gc}+1)} \tag{16}$$

This *log*-stabilized z-score transformation is the default normalization method in our R package called `Piccolo` (see Appendix for a discussion on other variance stabilization approaches implemented in `Piccolo`). Hereafter, we refer to it as the `Piccolo` normalization.

First-order approximations for estimates of mean and variance under variance stabilization transformation are valid for majority of the genes

The first-order approximations for the mean and variance of the counts under the variance stabilization transformation in our residuals-based normalization are based on the assumption that the transformed values (raw counts transformed by the variance stabilizing transformation) vary linearly over the range of raw counts close to the mean expression level. We tested the validity of this assumption for each gene by fitting a straight line (using linear regression) through the *log*-transformed values corresponding to those raw counts that fall within one standard deviation away from the mean expression value. We relied on the resultant adjusted R^2 (coefficient of determination) values to evaluate the validity of the assumption since values of adjusted R^2 close to 1 indicate that the transformed values can be approximated to lie along a straight line over the corresponding range of raw counts.

We found that for all UMI counts datasets included in our study, more than 85% of the genes had adjusted R^2 values greater than 0.8 (see Additional file 1: Table 1 and Fig. S18). Thus, for UMI counts data obtained using high-throughput protocols the first-order approximation utilized in our normalization method is valid, particularly for genes with small counts.

Piccolo normalization reduces the impact of sampling depth differences between cells while simultaneously ensuring variance stabilization

As stated earlier, the objective of normalization is to reduce or eliminate the systematic biases in counts between cells that are not reflective of actual biological differences. Since technical control data do not have any biological source of variation, differences in sampling depths are expected to be the major source of variation between the cells (droplets). In terms of PCA, this would translate to sampling depth showing up as a major contributor in one of the first few principal components (PCs).

In Fig. 3A, we show the scatter plots of cells based on their coordinates along PC1 and PC2. The colors of the dots reflect the size factors of the respective cells, with brighter shades (yellow) indicating larger size factors. Recall that larger size factors correspond to cells with larger sampling depths across the stable genes (see Eq. (7)). In order to evaluate whether size factors correlate with the first few PCs, we calculated the canonical correlation coefficient (ρ) [31] between the size factors and the top 5 PCs; ρ close to 1 would indicate strong correlation between the size factors and one of the top 5 PCs. For `Svensson 1` raw counts (left panel Fig. 3A), we can clearly observe a color gradient along PC1, with cells with larger size factors lying predominantly on the left and cells with smaller size factors predominantly on the right. Thus for raw counts, sampling depth differences indeed show up as a major source of variation between the cells. The value of the canonical correlation coefficient - $\rho = 0.97$ - supports this further.

Next, we applied the `Piccolo` normalization to `Svensson 1` (right panel in Fig. 3A) and confirmed that the sampling depth differences are no longer identified as a major source of variation by PCA. In fact, not only do we not observe a color gradient along PC1 or PC2, but even the canonical correlation coefficient between the size factors and the top 5 PCs is significantly reduced to $\rho = 0.3$ which suggests a weak correlation. Similar observations were made for another technical control data set [32] (see Fig. S17).

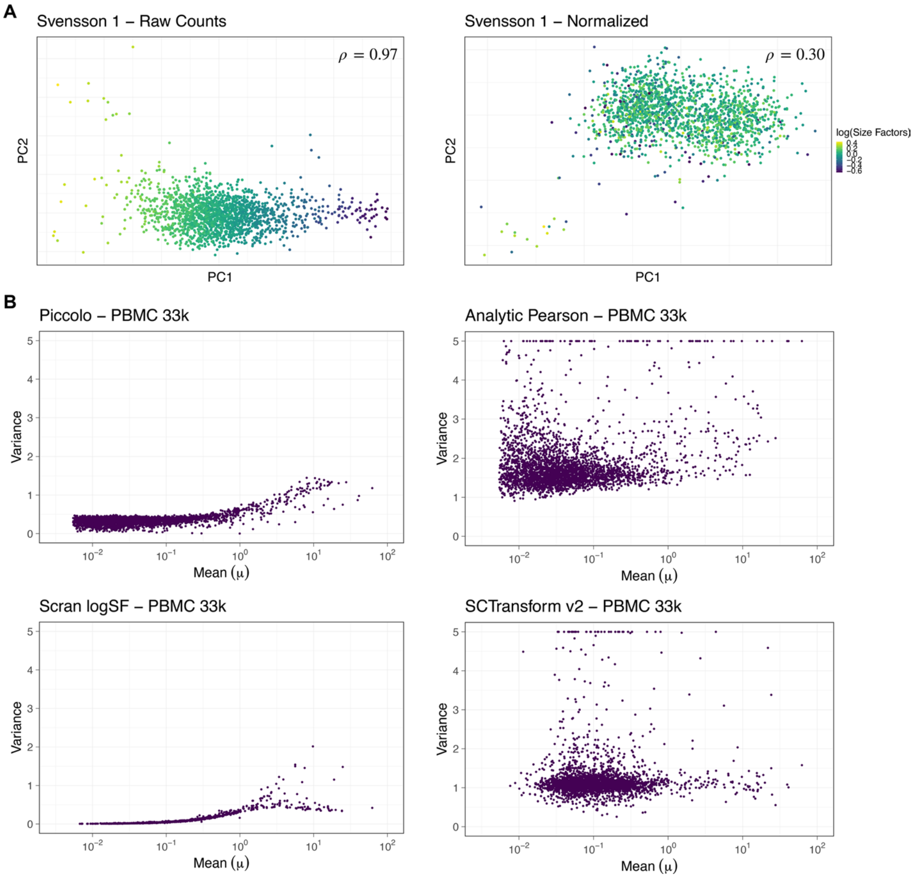

Fig. 3 Piccolo normalization reduces sampling depth differences between cells and also ensures effective variance stabilization. **A** 2-dimensional (2D) scatter plots based on the first 2 PCs of the Svensson 1 technical control data set. Each dot is a cell and is colored according to the size factors; brighter shades (yellow) correspond to larger size factors and darker shades (deep blue) correspond to smaller size factors. The left panel shows the 2D PC scatter plot for the raw counts, while the right panel shows the 2D PC scatter plot for the *z*-scores (residuals) obtained from Piccolo normalization. The coefficient (ρ) in the top-right corner of the panels shows the canonical correlation coefficient between the size factors and the top 5 PCs. Smaller values of ρ indicate that the impact of the sampling depth differences has been reduced more effectively by the normalization. Piccolo normalization reduces the impact of sampling depth on the overall variation as evaluated through PCA. **B** Variance vs mean (μ) linear-log scatter plots for the top 3000 HVGs of the PBMC 33k data set after applying respective normalizations - Piccolo (top-left panel), Analytic Pearson residuals (top-right), Scran logSF (bottom-left), and SCTransform v2 (bottom-right). The *y*-axis scale was limited to a maximum value of 5 to aid visual comparison; genes with variance greater than 5 were clipped to have the maximum value of 5. The residuals obtained using Piccolo exhibit variances that do not vary much beyond 1 unlike the the residuals obtained with Analytic Pearson and SCTransform v2

These results demonstrate that the `Piccolo` normalization is able to reduce the impact of systematic differences in sampling depths.

To examine the effectiveness of variance stabilization, we looked at the variances of the residuals after applying normalization to the raw counts. We compared `Piccolo` with two other residuals-based normalization methods - analytic Pearson [21](`Analytic Pearson`), and the regularized NB regression approach in scTransform [20, 29] (`SCTransform v2`). For reference, we also looked at the simple `logSF` based normalization approach implemented in Scran [12] (`Scran logSF`). Note, Scran relies on pooling of cells to arrive at estimates for the cell-specific size factors that are then used for the `logSF` normalization.

For the `PBMC 33k` data set, we first used our feature selection method to shortlist the top 3000 HVGs, and then applied the `Piccolo` normalization to compute the residuals corresponding to the raw counts of these HVGs. The `Analytic Pearson` residuals were also computed for these top 3000 HVGs identified with our feature selection method. This enables a direct comparison between the two methods since the residuals were calculated using the same set of features. In contrast, for `SCTransform v2` and `Scran logSF`, the top 3000 HVGs were shortlisted using their own respective approaches. For each of the normalization methods, we then calculated the variances of the residuals. These variances are shown in the variance-mean linear-log plots in Fig. 3B. It is apparent that the residuals obtained from `Piccolo` (Fig. 3B, top-left panel) exhibit much lesser scatter compared to the other two residuals-based approaches (Fig. 3B, top-right panel and bottom-right panel). In Fig. 3B bottom-left panel, we also show the variance of the *log*-transformed normalized values obtained with `Scran logSF` for reference. We note that the log-transformed values also exhibit much lesser deviation from 1 compared to the raw counts based residuals methods. Note, given the heteroskedastic nature of our counts data the increase in the variances of the transformed values (obtained from log-transformation based normalization or our variance stabilized residuals based normalization) as the mean expression levels increase is not eliminated. However, compared to the raw counts this dependence is reduced for the transformed values which plays an important role when we employ PCA downstream.

Piccolo feature selection and normalization lead to concrete improvements in downstream cell clustering

Residuals-based normalization which includes a variance stabilization transformation preserves cell-cell similarities between cells that share cell-type identities

Normalization is typically followed by dimensionality reduction and unsupervised clustering to identify groups of cells with similar expression profiles. Depending on the biological system, the groups of cells may correspond to distinct cell-types, or states. The identification of such groups is a pivotal step since it informs crucial downstream analyses such as differential expression and marker genes identification. The most popular scRNA-seq workflows (for example, Seurat [15–18] and Scanpy [19]) employ PCA to perform dimensionality reduction. Based on the PCs, *k*-nearest neighbour (*k*-NN) graphs are generated (with cells as nodes) in which communities of cells that are most similar to each other are detected using graph-partitioning algorithms such as Leiden [33] and Louvain [34].

To investigate how well our normalization method preserves cell-cell similarities between cells that share cell-type identities, we began by examining a truth-known data set (data set in which the cell-type identities of the cells are already known) prepared by Duo et al. [35] using cells purified with cell-type specific isolation kits by Zheng et al. [36]. Briefly, they prepared the data set by shortlisting purified cells belonging to 8 PBMC cell-types - B-cells, CD14 monocytes, CD56 NK cells, CD4 T-helper cells, memory T-cells, naive T-cells, naive cytotoxic T-cells, and regulatory T-cells - such that there were roughly equal numbers of cells corresponding to each cell-type in the final data set (between 400-600 cells per cell-type). We refer to this data set as `Zheng Mix 8eq`. Since `Zheng Mix 8eq` consists of a mix of well-separated cell-types (for instance, B-cells vs T-cells) and similar cell-types (different types of T-cells), it provides a simple yet reasonably challenging scenario for evaluating how well cells belonging to the different cell-types can be distinguished after normalizing the counts with the respective normalization methods.

We used `Piccolo` to shortlist the top 3000 HVGs and applied our normalization to obtain the residuals for those HVGs. For `Analytic Pearson`, the residuals were computed for the top 3000 HVGs shortlisted using our feature selection method, while for `Scran logSF` and `SCTransform v2` the transformed counts and residuals were respectively computed for the top 3000 HVGs shortlisted using their own methods. Subsequently, we performed PCA and shortlisted the first 50 PCs. We then used a simple *k*-NN based classification approach based on the PCs to predict cell-type labels for each cell by relying on the known cell-type labels (see Methods). Finally, we evaluated the extent of the agreement between the predicted labels and the known labels by calculating the following clustering metrics: the Macro F1 score, the adjusted Rand index (ARI), and the adjusted mutual information (AMI).

In Fig. 4A, we show the Uniform Manifold Approximation and Projection (UMAP) [37] plots for `Zheng Mix 8eq` with `Piccolo` normalization (top-left panel), `Analytic Pearson` (top-right panel), `Scran logSF` (bottom-left panel), and `SCTransform v2` (bottom-right panel). In the plots, each dot represents a cell and is colored according to the known cell-type labels (legend provided at the bottom of the 4 panels). Qualitatively, it is apparent from the UMAP plots that the B-cells, CD56 NK cells, and CD14 monocytes can easily be distinguished compared to the rest. As expected, it's more difficult to distinguish between the different kinds of T-cells, with memory and naive cytotoxic T-cells being the only ones that are comparatively easier to distinguish, particularly with the residuals-based approaches. The values of ARI, AMI, and Macro F1 quantifying the extent of the agreement between the predicted and the known cell labels are listed in the bottom-right corner of the panels for the respective normalization methods. While there isn't a significant difference in the metrics between the 4 normalization methods, we do observe the highest values with `Piccolo`.

However, these observations are not sufficient to argue for or against any of the methods. For a more robust comparison between the normalization methods, we created 100 subsets of `Zheng Mix 8eq` by randomly picking 50% of the cells in the original data set 100 times. For each of the 100 subsets, we used the same approach as discussed above for the 4 normalization methods and computed the respective ARI, AMI, and Macro F1 scores for the predicted cell labels based on the kNN-based classification approach. In Fig. 4B, we show the violin-box plots for the ARI, AMI, and Macro F1 scores for the 100 subsets. The colors correspond to the respective normalization methods used (red corresponds to `Piccolo`). We can clearly see that for all 3 clustering metrics, the highest values were consistently obtained with `Piccolo` normalization, reflecting that the best agreement between predicted and known labels is achieved using our feature selection and normalization method. For each clustering metric, we used paired Wilcoxon tests to quantify whether the differences between the values of the metric obtained with `Piccolo` normalization and other normalization methods were statistically significant

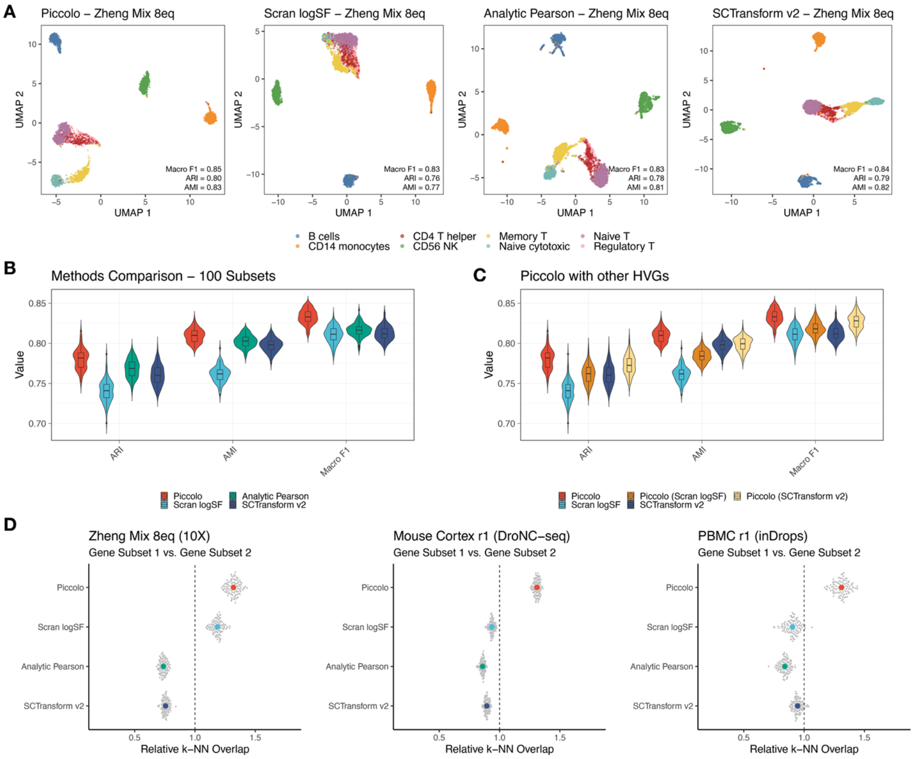

Fig. 4 Piccolo normalization preserves cell-cell similarities between cells sharing cell-type identities. **A** 2D UMAP plots after applying respective normalizations for the top 3000 HVGs of the Zheng Mix 8eq data set (consists of 3994 cells with roughly equal numbers of cells belonging to 8 distinct PBMC cell-types). Dots represent cells and are colored using the known cell-type labels (legend at the bottom of the panel). Clustering metrics - ARI, AMI, Macro F1 - based on comparisons between predicted cell labels (obtained using a kNN-based classification approach) and known cell labels are listed in the bottom-right corner in each panel. **B** Violin-box plots of the clustering metrics obtained for 100 subsets of the Zheng Mix 8eq data set. The colors correspond to the respective normalization methods used. For all the 100 subsets, the highest values of the metrics was observed with Piccolo (red) (seeAdditional file 1: Table 2). **C** Violin-box plots comparing the clustering metrics obtained for 100 subsets of the Zheng Mix 8eq data set with Piccolo normalization used with HVGs shortlisted obtained from other methods; Piccolo (Scran LogSF) and Piccolo (SCTransform v2) refer to the methods where the top 3000 HVGs were obtained using Scran logSF and SCTransform v2 respectively, and then the normalization was performed on those HVGs using Piccolo. Piccolo yielded higher values of the clustering metrics despite relying on HVGs shortlisted by other methods (see Additional file 1: Table 3). **D** Comparisons of overlap between the *k*-NN inferred separately on two halves of data split by genes. Relative *k*-NN overlap was calculated by dividing the mean overlap per data set by its average across all normalization methods. Colored dots indicate averages across the 100 splits (small grey dots) per normalization method - their colors are consistent with the colors in panel B. This panel is similar to Fig. 2a in [22] and highlights that Piccolo (red) surpasses the other methods in fulfilling the necessary condition of *k*-NN consistency

(see Methods). For all 3 metrics, the values obtained with `Piccolo` normalization were found to be consistently higher than those obtained with other methods (all paired Wilcoxon test p-values were found to be highly significant - $p < 1E - 11$, see Additional file 1: Table 2).

For our analyses on `Zheng Mix 8eq` so far, while `Piccolo` and `Analytic Pearson` normalization were applied on the same set of HVGs, the sets of HVGs for `Scran logSF` and `SCTransform v2` were different. Thus, some of the differences in the results are attributable to the differences in the sets of HVGs. To compare just the normalization methods, we used the top 3000 HVGs shortlisted by `Scran logSF` and `SCTransform v2` respectively, and computed the residuals using `Piccolo` normalization for these respective HVGs. `Piccolo (Scran logSF)` denotes the method wherein the HVGs were obtained from `Scran logSF` and then the `Piccolo` normalization was performed for those HVGs. Similarly, `Piccolo (SCTransform v2)` denotes the method wherein the HVGs were obtained from `SCTransform v2` and then the `Piccolo` normalization was performed for those HVGs. By applying these methods, we arrived at clustering metrics for the 100 subsets which are shown using the violin-box plots in Fig. 4C. The colors correspond to the feature selection and normalization methods used. We used paired Wilcoxon tests to compare the values of the metrics obtained with the normalization method that was used to shortlist the HVGs, with values of the metrics obtained with `Piccolo` normalization using those HVGs. For ARI and Macro F1, the values obtained with `Piccolo` normalization (`Piccolo (Scran logSF)` and `Piccolo (SCTransform v2)`) were consistently higher than the values obtained with the respective normalization methods with which the HVGs were shortlisted (paired Wilcoxon test $p < 1E - 09$). For AMI, while the values with `Piccolo (Scran logSF)` were consistently higher than with `Scran logSF` (paired Wilcoxon test $p < 1E - 17$), the values obtained with `Piccolo (SCTransform v2)` were not as significantly high compared to the ones obtained with `SCTransform v2` (paired Wilcoxon test $p < 0.06$) (see Additional file 1: Table 3). Overall, it is clear that even with other HVGs, `Piccolo` normalization is better at preserving cell-cell similarities between cells of the same cell-type compared to the other normalization methods. Moreover, from Fig. 4C what is most striking is that the values of the clustering metrics obtained using our feature selection and normalization are the highest overall (red violin-box plot corresponds to `Piccolo`). This clearly suggests that not only is the performance of `Piccolo` normalization consistently better than the other normalization approaches (assessed with the aid of the clustering metrics), but even our feature selection method is effective at shortlisting HVGs that better inform the differences between cells with distinct cell-type identities.

Piccolo maintains consistency of k-nearest neighbors - a necessity for ensuring robustness of downstream analyses involving nearest neighbor graphs

In a recent article comparing different transformations for scRNA-seq data, Ahlmann-Eltze and Huber highlighted the k-nearest neighbor (k-NN) graph as a fundamental data structure which is used to infer cell-types, states, trajectories etc [22]; to remind, the k-NN graph in scRNA-seq analyses is obtained by relying on a lower-dimensional representation of the cells using PCA, and subsequently shortlisting the k-NNs of each cell based on Euclidean distances in the PC space (typically k is 10). They pointed out that the consistency of the k-NNs is a necessary (albeit not sufficient) condition for the robustness of downstream analyses that rely on k-NN graphs. They evaluated k-NN consistency by evenly splitting each data set into two halves based on the genes, such that the two resultant subsets contained mutually exclusive sets of genes (they referred to them as gene subset 1 and gene subset 2, see Fig. 2a in [22]). They then applied the respective transformation approaches to the gene subset 1 and gene subset 2 datasets separately and obtained k-NNs for each cell corresponding to each of the subsets. Using these 2 sets of k-NN cells for each cell, they performed a pairwise comparison to determine the extent of overlap between them and thereby assess consistency.

We performed a similar analysis for the `Zheng Mix 8eq` (10X), `Mouse Cortex r1` (DroNC-seq), and `PBMC r1` (inDrops) datasets (see Methods). While Ahlmann-Eltze and Huber only considered 10X derived UMI counts datasets, we included datasets obtained from other droplet-based high-throughput technologies to showcase the performance of Piccolo in preserving *k*-NN consistency. We performed the gene-based splits for each data set 100 times. *k*-NN overlaps were obtained per cell and then averaged across all the cells to arrive at one mean estimate per iteration. Relative *k*-NN overlap values were calculated by dividing these mean NN overlap values by their average across all iterations for all 4 normalization methods. Figure 4D shows the resultant values of the relative *k*-NN overlaps for each normalization method for the 100 splits (small grey dots). The large colored dots indicate the averages across the 100 splits per normalization method; colors of the dots were kept consistent with the colors used in panel B for each of the methods. Unlike Fig. 2a in [22] where the authors aggregated the relative *k*-NN overlap values across the datasets, we show the relative *k*-NN overlaps for each data set separately to highlight that Piccolo (red) easily surpasses the other methods in fulfilling the necessary condition of *k*-NN consistency despite basic differences in these droplet-based high-throughput technologies (all paired Wilcoxon test *p*-values between Piccolo and other methods were less than $2.2E-10$). Based on these results, we conclude that our proposed normalization method ensures the consistency of *k*-NNs and will therefore enable more robust inferences to be drawn from downstream analyses that rely on *k*-NN graphs.

Piccolo enables identification of groups containing few cells as well as groups with cells that express fewer differentially expressed genes

To further investigate the performance of `Piccolo`, we utilized 3 simulation tools—SPARSim [38], Splat [39], and SCRIP [40]— to simulate counts based on estimation of parameters derived from real UMI counts datasets. We selected these 3 tools since they enable *de novo* simulations of single cell counts wherein we can specify both the number of groups of cells, as well as the extent of differential expression between the groups. In a recent benchmarking study of scRNA-seq simulation methods [41], SPARSim was in fact found to rank among the best in terms of overall performance (see Fig. 2 in [41]). In order to generate simulated counts for user-specified number of groups, all 3 methods require a homogeneous cell population as input to estimate the parameters for simulation. Keeping this in mind, we prepared a data set consisting exclusively of B cells obtained from the `Zheng Mix 8eq` data set. In addition, we used the `NIH/3T3` and `HEK293T` datasets since the cells in these two datasets are derived from cell lines and therefore constitute homogeneous cell populations.

We explored the following two simple simulation scenarios to benchmark and compare the normalization methods:

- `Scenario 1`: 6 groups of cells with different number of cells per group, while keeping the extent of differential expression between the groups roughly the same. The objective behind this simulation was to examine whether the groups with the fewest cells can be reliably identified after applying the different normalization methods.

- `Scenario 2`: 6 groups of cells with the same number of cells per group, but with different numbers of differentially expressed genes in each group. The objective behind this simulation was to examine how well we can distinguish between cells belonging to distinct groups, especially the ones that have fewer differentially expressed genes.

For each scenario and for each of the 3 reference datasets, we simulated 50 datasets using the 3 simulation tools respectively (see Methods for details). This gave us 450 simulated counts datasets for each scenario discussed above (900 in total). For each data set, we applied the 4 normalization methods and applied PCA to the residuals/transformed counts to shortlist the first 50 PCs. This was followed by the application of our *k*-NN based classification approach to predict the labels for each cell based on the known cell-group labels. We then quantified the agreement between the predicted labels and the known labels by estimating ARI, AMI, and Macro F1 values. In panel A in Fig. 5, we show the violin-box plots for the 3 clustering metrics for the simulated datasets generated for `Scenario 1` (see Additional file 1: Fig. S22 for the corresponding panels for `Scenario 2`). The colors correspond to the normalization methods used. The panels in Fig. 5A are arranged such that the rows correspond to the reference datasets and the columns correspond to the simulation tool that was used to simulate the 50 datasets. We used paired Wilcoxon tests to compare the values of the metrics obtained with other normalization methods with those obtained after applying `Piccolo`. The results are summarized in Additional file 1: Tables 4, 5, 6, 7, 8 and 9. Overall, we observed that regardless of the data set or the simulation tool used to generate the simulated counts, the clustering metrics yielded after applying `Piccolo` (red) were consistently among the highest. Only for the simulated counts datasets generated by SPARSim for `Scenario 2` based on the `NIH/3T3` and `HEK293T` datasets were the clustering metrics obtained with `Scran LogSF` marginally but consistently higher (see Additional file 1: Fig. S22 and Table 5). In stark contrast, for the simulated counts datasets generated by Splat for both `Scenario 1` and `Scenario 2` based on the `NIH/3T3` and `HEK293T` datasets the clustering metrics obtained with `Scran LogSF` were consistently the lowest, while with `Piccolo` we continued to observe the highest values.

In order to evaluate how well the methods enabled the identification of rarer cell populations (Group 6 in `Scenario 1`), as well as groups of cells with fewer differentially expressed genes (Group 6 in `Scenario 2`), we aggregated the group-wise F1 scores across all the simulated datasets for each scenario. Panels B and C show the heatmaps of the mean F1 scores per group for `Scenario 1` and `Scenario 2` respectively. The tiles of the heatmap have been colored based on the values of the F1 scores, with brighter shades (yellow) indicating larger values (greater agreement between the predicted and the known group labels), and darker shades (dark blue) indicating smaller values (less agreement between the predicted and the known group labels). For both scenarios, the highest mean F1 scores for all the groups were obtained after using `Piccolo`. Note in particular, the differences between the mean F1 scores for Group 6 for `Scenario 1`. The fact that these improvements are so noticeable after aggregating across 450 datasets underscores the concrete improvements in cell-cell grouping enabled by our feature selection and normalization method.

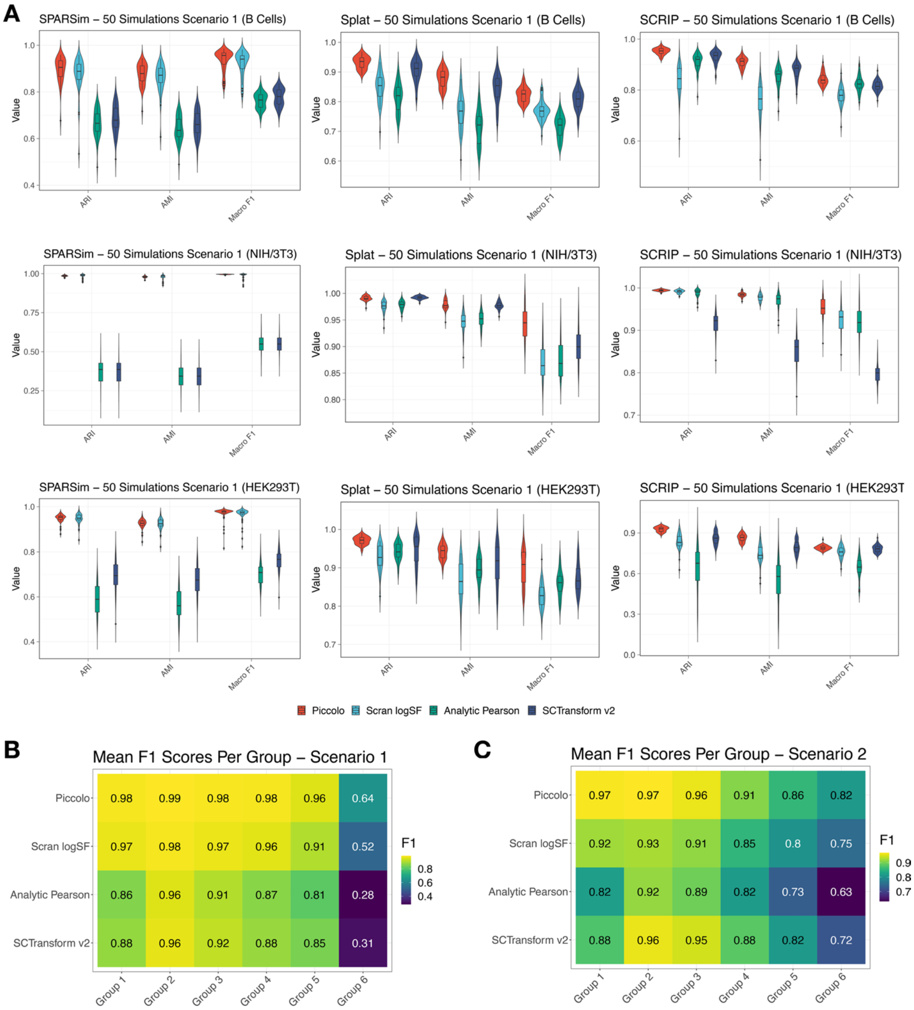

Fig. 5 Piccolo enables identification of groups containing fewer cells as well as groups of cells with fewer differentially expressed genes. **A** Violin-box plots of the clustering metrics—ARI, AMI, Macro F1—obtained for simulated counts data generated for Scenario 1 based on parameters derived from 3 different real datasets (each row corresponds to one data set) and 3 different simulation tools (each column corresponds to one simulation tool). The colors correspond to the respective normalization methods used. For most simulated counts datasets, the highest values of the metrics were observed with Piccolo (red). **B** Heatmap showing the mean F1 scores per group aggregated over all the simulated counts data for Scenario 1 for each of the normalization methods. The tiles of the heatmap are colored according to the mean F1 scores with larger scores corresponding to brighter shades (yellow), and lower scores corresponding to darker shades (deep blue). Piccolo has the highest mean F1 scores for all 6 groups, with the improvement especially notable for Group 6 (group with fewest cells). **C** Same as panel B but for Scenario 2 (see Additional file 1: Fig. S22). Piccolo has the highest mean F1 scores for all 6 groups. Note, colors are scaled differently in panels B and C to bring out the differences between the mean F1 scores in the respective simulation scenarios

The consistency with which we observed higher values of the clustering metrics with `Piccolo` becomes all the more striking when we take into account the differences in the nature of the counts data generated by the different simulation frameworks. It is well understood that all simulation frameworks have limitations, and cannot faithfully capture all the attributes of real datasets [41]. Overall, SPARSim generated counts that appeared to closely follow the expected mean-variance relationship (see σ^2 vs. μ log-log and α_{QP} vs. μ log-log plots in Additional file 1: Figs. S4, S5 and S6). SCRIP generated simulated counts data exhibited the most unusual mean-variance behavior, especially for genes with larger μ (see Additional file 1: Figs. S10, S11 and S12). This was surprising considering that it was proposed as an improvement over the popular Splat framework. Given that our normalization method relies on first-order approximations for the mean and variance of the *log*-transformed counts, we expected that such differences in the nature of the counts would lead to significant variations in the performance of `Piccolo` and potentially lead to poor results. However, as demonstrated with the aid of these sets of simulations the method is robust to such differences as long as they don't deviate too significantly from the expected behavior for counts usually obtained from high-throughput protocols.

In summary, with the help of `Piccolo` we observed concrete improvements in the clustering outputs, both in terms of identification of rarer cell groups as well as groups of cells that expressed fewer DE genes. The simulation scenarios were deliberately kept simple in order to facilitate a more straightforward comparison and interpretation. When we factor in the differences in the nature of counts generated by the different simulation tools as well as the fact that the counts were generated based on parameters derived from 3 independent datasets, the robustness of the improvements enabled by `Piccolo` become all the more compelling.

Discussion

We began our investigation into the nature of UMI counts by examining the mean-variance relationships of the observed counts for the genes and showed that for genes with small counts the variance of the counts can be approximated quite well by the quasi-Poisson variance. We pointed out that this quasi-Poisson nature of the variance simply reflects the fact that the counts for the respective genes are small in most cells. We followed this by examining and questioning the assumption underlying typical scRNA-seq workflows. In a typical scRNA-seq workflow, feature selection is preceded by normalization. Implicit in this sequence of steps is the assumption that features which exhibit high biological variability can be identified only after taking into account the systematic technical biases. We pointed out that this assumption reflects a confusion between the distinct objectives of identification of the differentially expressed genes, and the HVGs. While it is imperative that the counts be normalized to account for the systematic biases prior to a differential expression analysis, we showed that in fact it is possible to identify HVGs based on just the observed counts. We proposed a simple approach for feature selection that relies on quasi-Poisson dispersion coefficients estimated from the observed counts using a regression-based method. A key advantage with assessing the overall variability of counts for each gene prior to normalization is the ability to identify genes whose counts do not vary significantly across the cells. We refer to these genes as *stable genes*. We posited that the variability of counts for such stable genes is primarily reflective of the systematic biases (such as sampling depth differences) and can more reliably inform the estimation of size factors for each cell. Based on these observations, we propose a revision of the scRNA-seq workflow in which feature selection precedes, and in fact informs normalization (see Fig. 6).

Before proceeding to discuss the salient aspects of our revised workflow and the normalization method proposed in this paper, it will be helpful to summarize and highlight

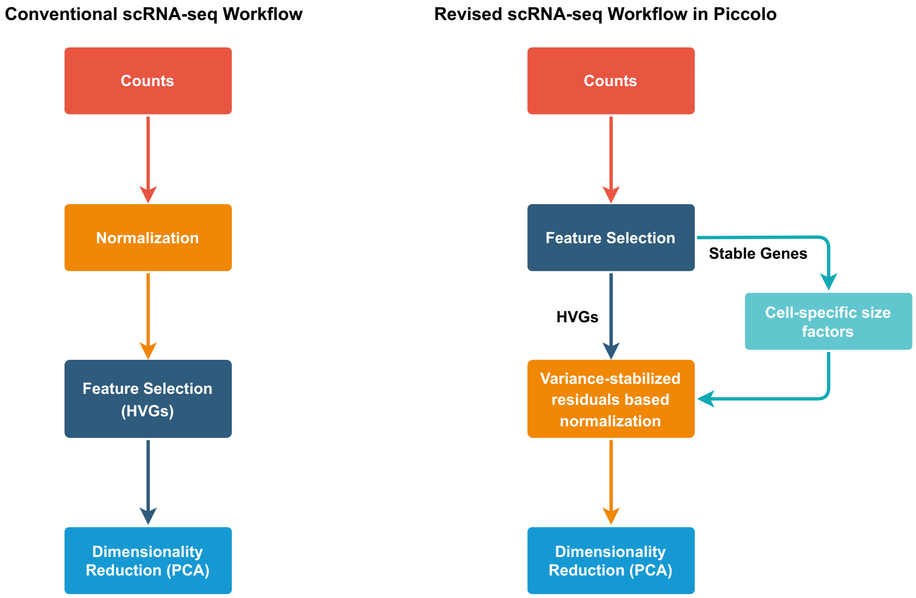
Conventional scRNA-seq Workflow
Counts
Normalization
Feature Selection (HVGs)
Dimensionality Reduction (PCA)
Revised scRNA-seq Workflow in Piccolo
Counts
Feature Selection
Stable Genes
Cell-specific size factors
HVGs
Variance-stabilized residuals based normalization
Dimensionality Reduction (PCA)

Fig. 6 The revised scRNA-seq workflow implemented in Piccolo. We contrast the conventional scRNA-seq workflow (left) with the revised scRNA-seq workflow (right) proposed by us and implemented in Piccolo. The starting point for all analyses are the counts matrices. In the conventional workflow, the counts are first normalized and variance stabilized followed by feature selection to identify HVGs. Dimensionality reduction is then performed on the normalized counts. In contrast, in the revised workflow proposed by us, we first perform feature selection to identify both the HVGs as well as stable genes. We rely on the stable genes to estimate cell-specific size factors for performing normalization. During normalization, we also ensure variance stabilization which in turn leads to significant improvement while performing dimensionality reduction with PCA

some key aspects of the existing methods. In a recent article comparing different transformations for scRNA-seq data, Ahlmann-Eltze and Huber showed that the simple `logSF` normalization outperformed the residuals-based normalization methods particularly in ensuring consistency of the *k*-nearest neighbors [22]; consistency was evaluated by evenly splitting the data into two halves such that the two subsets contained mutually exclusive sets of genes, and inquiring whether the cells shared the same sets of neighbouring cells between the split subsets (nearest neighbour cells were identified based on Euclidean distances between the cells in the PC space). This consistency is primarily attributable to the variance stabilization ensured by the log-transformation. However, as pointed out earlier, a significant disadvantage of the `logSF` normalization is that it is not very effective at reducing the sampling depth differences between the cells (Additional file 1: Fig. S15). In contrast, the residuals-based methods reduce sampling-depth differences between the cells much more effectively (Additional file 1: Fig. S16). However, a drawback of the residuals-based approaches is that they are unable to effectively reduce and stabilize the variances, particularly for marker genes (see Appendix and [22]).

Keeping in mind the respective advantages and disadvantages of the `logSF` normalization and the residuals-based methods described above, we proposed a residuals-based (*z*-scores) based normalization method which includes a variance stabilization transformation (the default transformation function used in `Piccolo` is the *log*, see Appendix for descriptions of other options).

Given the observation that the majority of counts obtained from high-throughput technologies are small, we relied on first-order approximations for the means and variances of the transformed counts to estimate the residuals (see Additional file 1: Fig. S18). In addition, the residuals relied on means and variances that were adjusted using the size factors estimated with the help of stable genes. Thus, our conceptually straightforward approach simultaneously ensures variance stabilization and reduces the impact of the sampling depth differences thereby leading to concrete improvements in the downstream clustering performance.

We first applied our normalization method on technical control datasets (Fig. 3A and Additional file 1: Fig. S17) and demonstrated that it does reduce the impact of sampling depth differences between the cells. Using the PBMC 33k data set (and other datasets, see Additional file 1: Fig. S19), we showed that our normalization method also ensures effective variance stabilization. We then applied `Piccolo` to a truth-known data set, and showed that it improves downstream clustering analysis and is able to preserve cell-cell similarities between cells that share cell-type identities better than other popular normalization methods. We also examined how well the consistency of *k*-NNs was ensured by `Piccolo`, and demonstrated that it surpasses the other methods in satisfying this basic requirement to ensure the robustness of downstream analyses that rely on *k*-NN graphs. With the help of simulations, we were also able to show that `Piccolo` consistently enabled the identification of groups that contain small number of cells, whereas other normalization methods failed to consistently and reliably ensure the same. In addition, we also showed that with `Piccolo` we can better distinguish between groups that express fewer differentially expressed genes. These results are especially relevant biologically when viewed from the perspective of the identification of rare cell-types, or distinguishing between cell states.

While our method offers significant improvements over the existing workflows, we now discuss some of its limitations. Beginning with feature selection, a fundamental assumption underlying our bin-based approach is that across all expression levels there are always some genes which are not biologically variable. This assumption is not motivated by biological observations and has been made to facilitate ease of computation. It is possible that we overlook some HVGs because of this assumption. In our current implementation, we mitigate this by keeping the level for the reference dispersion coefficient in each bin relatively low (10th quantile is the default). However, there is definite scope for further improvement of the feature selection process to ensure that the HVGs are more effectively identified. Another point of consideration tied to feature selection is the use of stable genes for estimating cell-specific size factors. Due to the sparsity of the data, it is possible that the counts across the stable genes for some cells are all zero resulting in the size factor estimates to be zero for those cells. In `Piccolo`, we address this by iteratively adding sets of genes from the bottom of the list of HVGs till none of the cells have a size factor estimate of zero. Despite this limitation, we still expect that these size factors will not be confounded by actual biological variation as much as when we estimate them using all the genes (since that includes the HVGs). With regard to our normalization method, we relied on the first-order approximations for both the mean and the variance under the variance stabilization transformation. These approximations will work well as long as the non-linear transformation function is approximately linear in the range of the observed counts (see Additional file 1: Fig. S18). For small counts, this is indeed true, however for larger counts these approximations may lead to incorrect estimates. Given the nature of the droplet-based UMI counts data at present, our results suggest that the first-order approximations work quite well. We point out here that for datasets that exhibit larger counts and less sparsity, the conventional approach of the log-based normalization can be expected to work reliably and is available as one of the options in the Piccolo R package. We also want to note that in this study we did not discuss and elaborate on differential expression (DE) analysis which forms a vital component of all scRNA-seq studies. This was done to focus attention on the conceptual clarifications and simplifications for the core steps of feature selection and normalization that shape all downstream analyses, including identification of DE genes. We remind here that if we are unable to consistently and reliably identify groups of cells that actually correspond to distinct cell-types or states, then the downstream DE analyses are unlikely to be as informative and helpful. In the worst cases, they may even be misleading. Briefly, we would like to mention here that after applying our normalization method to the observed counts, the distribution of the residuals are brought closer to the normal distribution, thus making it possible to employ the two-sample Student's t-test with the null hypothesis that the means of the two samples are the same. Typically, in scRNA-seq analyses the Wilcoxon rank-sum test, which is a non-parametric alternative to the two-sample t-test, is the preferred choice.

Another point of consideration for single-cell workflows is the time taken to perform the normalization, as well as the memory (RAM) that is used. For the latter, since our residuals-based normalization transforms a sparse matrix to a dense matrix, the amount of memory that is used for the post-normalization matrix will increase significantly. However, since we can shortlist the HVGs before normalization, the transformed counts matrix will only be generated for the HVGs thus requiring lesser memory than what would be needed if normalization was performed for all genes. With regards to the computation time, since the transformation is based on a simple analytical relation, it only took between 15 seconds (for `Svensson 1`) to up to 2.5 minutes (for `PBMC 33k`) to transform the observed counts to the residuals for each of the datasets used in this paper (see Methods for system configuration).

In conclusion, the novel scRNA-seq workflow based on the conceptual simplifications presented in this article enable consistent and significant improvements in the downstream analyses. We expect that with the aid of the implementation of this workflow in `Piccolo` more cogent and impactful inferences will be drawn from future single-cell gene expression studies.

Methods

Data preparation and preprocessing

The datasets used in the this study are listed in the table in the Datasets section, along with the links to the sources.

Cell and gene filtering

Primarily, the only cell filtering applied for all datasets was to ensure that all the cells had non-zero total counts. However for the evaluation of the effectiveness of our normalization method using PCA, we filtered cells from the `Svensson 1` data set that had total counts 3.5 median absolute deviation away from the median total count. This was done to reduce the impact of outliers on PCA.

For all the datasets, we excluded genes that had fewer than 0.5% cells with non-zero counts. This is the default gene filtering employed in `Piccolo`.

Selection of HVGs

For all the analyses, unless specified otherwise, we shortlisted the top 3000 HVGs. For `Piccolo` and `Analytic Pearson` residuals-based normalization, the identification of HVGs was done using our dispersion coefficient-based feature selection method. For `Scran logSF` and `SCTransform v2`, the top 3000 HVGs were shortlisted using their respective approaches that rely on post-normalization transformed counts/residuals to identify the genes with largest variances of the transformed counts/residuals.

Dimensionality reduction using PCA

After selecting the top 3000 HVGs and performing normalization on the counts of the HVGs, we used PCA for dimensionality reduction and shortlisted the top 50 PCs. Prior to using PCA, for the residuals-based methods (including `Piccolo`) we centered the residuals at 0, but did not scaled to unit variance. For the `logSF` normalization, we did not center or scale the transformed counts.

Kendall's and Spearman's rank correlation tests

Kendall's and Spearman's rank correlation tests were used to evaluate whether there is a statistical dependence between the quasi-Poisson dispersion coefficient (α_{QP}) and the mean expression levels (μ) for genes with $\mu < 0.1$. Both tests evaluate how well the relationship between two variables can be described using a monotonic function. For both tests, the correlation coefficients - τ and ρ respectively - indicate a statistical dependence if the values are close to $+1$ or -1, while values of τ or ρ closer to 0 indicate the absence of such a statistical dependence. For these tests, genes with non-zero counts in fewer than 2.5% of cells in the respective datasets were not included.

Benchmarking cell-type clustering and separation using *k*-NN based classification

Our *k*-NN based approach to predict cell labels using known cell labels is based on a very simple premise. After we normalize the observed counts and perform PCA, the expectation is that in the PC space the cells that share cell-type (or group) identities are close to each other. Thus, if we examine the nearest neighbours of a given cell belonging to a given cell-type (or group), we expect to find that the nearest neighbours are predominantly cells belonging to the same cell-type (or group). Based on this simple expectation, we predict cell labels for each cell by considering its k (default $k = 15$) nearest neighbours and using the known cell-type (or group) labels to identify the cell-type (or group) that is most over-represented in the nearest neighbour set - the most over-represented cell-type is the predicted cell-type label for the given cell. Over-representation is assessed using the hypergeometric test. By testing for over-representation, we ensure that there is no bias against cell-types (or groups) that have fewer cells while predicting the cell-type identity for any given cell.

Comparing clustering metrics obtained using the different normalization methods

For the 100 random subsets generated using the `Zheng Mix 8eq` data set, we applied the respective normalization methods and performed dimensionality reduction using PCA. We used the top 50 PCs for each of them, and with our kNN-based classification approach predicted cell-type labels for each cell that we then compared with the known cell labels using the following clustering metrics: Macro F1 (harmonic mean between precision and recall, averaged across the classes), adjusted Rand index (ARI), and adjusted mutual information (AMI). We compared the values of each of these metrics obtained with the different normalization methods by using the paired Wilcoxon rank-sum test. The null hypothesis is that there is no difference in the values of the metrics between the two groups being compared. The paired-test is essential in this context since the values of the metrics have to be compared pairwise for the same subset, and cannot be compared across different subsets.

The same 100 subsets were used to also evaluate how well `Piccolo` performs when we rely on HVGs shortlisted based on other normalization methods. Once again, we relied on the paired Wilcoxon rank-sum test to assess whether there are differences in clustering performance as evaluated through the clustering metrics: Macro F1, ARI, and AMI.

Evaluating *k*-nearest neighbors consistency

For the `Zheng Mix 8eq` (10X), `Mouse Cortex` r1 (DroNC-seq) and `PBMC r1` (inDrops) datasets, we randomly split the genes into two even subsetted datasets - Gene Subset 1 and Gene Subset 2. We normalized the observed counts in each subset using the four methods discussed in the paper. We performed PCA on the normalized values to shortlist the top 50 PCs and identified the 10 nearest neighbors of each cell by relying on Euclidean distances between pairs of cells in the 50 dimensional PC space (dbscan [42, 43] was used to identify the NN cells). For each cell, we thus obtained two sets of 10 nearest neighbors from the respective subsets and quantified the extent of their overlap. We then took the mean of these per cell overlaps across all cells in the given data set to calculate the mean overlap for any given normalization method.

This procedure was repeated 100 times and the mean overlap values were recorded for each iteration. To obtain the relative *k*-NN overlap values, we took the ratio of each of the mean overlap value obtained from a given iteration for each normalization method with the average of all the mean overlap values across all iterations and all normalization methods. This yielded the relative *k*-NN overlap values shown in panel D in Fig. 4.

Simulations

We used SPARSim [38], Splat [39], and SCRIP [40] to generate the simulated counts datasets. SPARSim scRNA-seq simulator is based on a Gamma-Multivariate Hypergeometric model, while Splat relies on the Gamma-Poisson distribution to simulate counts. Single-cell RNA sequencing information producer (SCRIP) extends the framework of

Splat to be capable of simulating data by relying on Gamma-Poisson or Beta-Gamma-Poisson models (latter specifically designed for modeling the transcriptional bursting effect). Splat and SCRIP are similar in the way the choices of parameters need to be specified for *de novo* simulations - we can specify both the number of groups, as well as the extent of differential expression between the groups (this is specified in terms of the probability that a gene will get picked to be differentially expressed in each group). In order to generate the simulated counts, they both require a homogeneous cell population based on which the parameters for the simulation are estimated. SPARSim also allows for the exact specification of the number of cells per group with simulation of multiple cell-types/groups accomplished by introducing DE genes through user-specified fold-changes for subsets of genes. Their procedure for the introduction of DE genes accounts for the well-known fact that gene biological variability is related to gene expression level, thus the fold-changes result in changes not only the mean gene expression level but also in gene variability values (implemented in their `SPARSim_create_DE_genes_parameter(sim_param, fc_multiplier)` function).

Below we provide details of the parameters specified for each of the simulation methods for the two simulation scenarios described in the main text:

- **SPARSim (Scenario 1):** First, we estimated the simulation parameters for each reference data set - B cells (`Zheng Mix 8eq`), `NIH/3T3`, and `HEK293T`. For each data set, we randomly picked 10000 genes to use for generation of the simulated counts. This was done 50 times to generate 50 simulated counts datasets corresponding to each of the reference data set. We adopted the suggestion of the authors of SPARSim to create multiple cell-types/groups by specifying fold-changes between the homogenous cell population used to estimate the simulation parameters (hereafter referred to as cell type A) and the new cell-type/group for a subset of genes; half of the DE genes were specified to have fold-changes between 0.05 to 0.1 (sampled uniformly) for down-regulation and the other half were specified to have fold-changes between 1.5 and 2 (sampled uniformly) for up-regulation compared to cell type A. For the B cells based simulation, we set the number of DE genes to be 430, while for `NIH/3T3` and `HEK293T` we set the number of DE genes to be 150 and 80 respectively. To simulate 6 groups overall, we picked 5 disjoint and random sets of DE genes to simulate 5 groups (apart from cell type A). The number of cells in each group were specified as follows: Group 1 (cell type A) - 750, Group 2 - 750, Group 3 - 400, Group 4 - 250, Group 5 - 250, Group 6 - 100. In total, we had 2500 cells in the simulated datasets.
- **Splat (Scenario 1):** For each reference data set (B cells (`Zheng Mix 8eq`), `NIH/3T3`, and `HEK293T`), we first randomly picked 10000 genes to use for estimation of parameters and generation of the simulated counts. This was done 50 times to generate 50 simulated counts datasets corresponding to each of the reference data set. We simulated 6 groups containing cells in the following proportions: Group 1 - 0.30, Group 2 - 0.30, Group 3 - 0.20, Group 4 - 0.15, Group 5 - 0.04, and Group 6 - 0.01. The probability that a gene will be picked to be differentially expressed was kept fixed at 0.4 for all the groups (parameter `de.prob` in Splatter).
- **SCRIP (Scenario 1):** For each reference data set (B cells (`Zheng Mix 8eq`), `NIH/3T3`, and `HEK293T`), we first randomly picked 10000 genes to use for estima-

tion of parameters and generation of the simulated counts. This was done 50 times to generate 50 simulated counts datasets corresponding to each of the reference data set. We simulated 6 groups containing cells in the following proportions: Group 1 - 0.30, Group 2 - 0.30, Group 3 - 0.20, Group 4 - 0.15, Group 5 - 0.04, and Group 6 - 0.01. The probability that a gene will be picked to be differentially expressed was kept fixed at 0.5 for all the groups (parameter `de.prob` in SCRIP).

- **SPARSim (Scenario 2):** First, we estimated the simulation parameters for each reference data set - B cells (`Zheng Mix 8eq`), `NIH/3T3`, and `HEK293T`. After estimating the parameters, for each data set we randomly picked 10000 genes to use for generation of the simulated counts. This was done 50 times to generate 50 simulated counts datasets corresponding to each of the reference data set. For the B cells based simulation, to simulate 6 groups overall we picked 5 disjoint and random sets of DE genes to simulate 5 groups (apart from cell type A) with the number of DE genes in each group specified as follows: Group 2 - 1750, Group 3 - 1400, Group 4 - 1050, Group 5 - 700, Group 6 - 350 (note Group 1 was the reference group). For `NIH/3T3` and `HEK293T`, the number of DE genes in each group were specified as follows: Group 2 - 250, Group 3 - 200, Group 4 - 150, Group 5 - 100, Group 6 - 50 (note Group 1 was the reference group). As with `Scenario 1`, the DE genes were specified to have fold-changes between 0.05 to 0.1 (sampled uniformly) for down-regulation and the other half were specified to have fold-changes between 1.5 and 2 (sampled uniformly) for up-regulation compared to cell type A. For all SPARSim based simulations of `Scenario 2`, the number of cells in each group were kept fixed at 400. In total, we had 2400 cells in the simulated datasets.
- **Splat (Scenario 2):** For each reference data set (B cells (`Zheng Mix 8eq`), `NIH/3T3`, and `HEK293T`), we first randomly picked 10000 genes to use for estimation of parameters and generation of the simulated counts. This was done 50 times to generate 50 simulated counts datasets corresponding to each of the reference data set. We simulated 6 groups containing roughly equal numbers of cells (total number of cells set to 2500). The probability that a gene will be picked to be differentially expressed per group was specified as follows: Group 1 - 0.25, Group 2 - 0.2, Group 3 - 0.15, Group 4 - 0.1, Group 5 - 0.05, Group 6 - 0.025.
- **SCRIP (Scenario 2):** For each reference data set (B cells (`Zheng Mix 8eq`), `NIH/3T3`, and `HEK293T`), we first randomly picked 10000 genes to use for estimation of parameters and generation of the simulated counts. This was done 50 times to generate 50 simulated counts datasets corresponding to each of the reference data set. We simulated 6 groups containing roughly equal numbers of cells (total number of cells set to 2500). The probability that a gene will be picked to be differentially expressed per group was specified as follows: Group 1 - 0.25, Group 2 - 0.2, Group 3 - 0.15, Group 4 - 0.1, Group 5 - 0.05, Group 6 - 0.025.

System configuration and software used

We used a Macbook Pro with the Apple M1 pro chip and 16GB RAM for all the analyses in this paper. `Piccolo` was developed using R (version 4.2.0), and all the analyses were also performed using R [44]. We used the following R packages in this study: cluster [45], data.table [46], dbscan [42, 43], ggplot2 [47], igraph [48], Matrix [49], matrixTests [50], RSpectra [51], Rtsne [52–54], umap [55], viridis [56]. In addition, the `Piccolo` package also includes a helper function made by Kamil Slowikowski called writeMMgz to help prepare .mtx.gz files.

Datasets

Name	Technology	Tissue	Type	Source	Reference
Svensson 1	10X Chromium v1	Technical Control	Technical Control	Link	[10, 57]
Klein 2015	inDrops	Technical Control	Technical Control	Link	[10, 32]
NIH/3T3	10X Chromium v3	3T3	Cell Line	Link	[10]
HEK293T	10X Chromium v3	HEK	Cell Line	Link	[10]
PBMC 33k	10X Chromium v1	PBMC	Heterogeneous	Link	[20]
PBMC r1	inDrops	PBMC	Heterogeneous	GSE132044	[8]
Mouse Cortex	DroNC-seq	Cortex	Heterogeneous	GSE132044	[8]
Mouse Lung	Drop-seq	Lung	Heterogeneous	GSE124872	[25]
Zheng Mix 8eq	10X Chromium v1	PBMC	Heterogeneous	Link	[35, 58]
Haber 2017	10X Chromium v1	Mouse Small Intestine Epithelium	Heterogeneous	GSE92332	[59]

Appendix

Regression-based approach to estimate QP dispersion coefficients

The regression-based test proposed by Cameron and Trivedi [23, 24] relies on the idea that under the null hypothesis - counts, x, are Poisson distributed - the expected value of $(x - E[x])^2 - x$ is zero, while under the alternative hypothesis the expected value would be a scalar multiple of a function of $E[x]$. Since $E[x]$ is unknown, we replace it by the estimate under the null hypothesis and estimate the scalar multiple using least squares regression.

The choice of the QP mean-variance relation corresponds to the alternative hypothesis in which the variance ($Var[x]$) is a linear function of the mean ($E[x]$),

$$Var[x] = \alpha_{QP}.E[x] \tag{17}$$

Given this specification of the alternative hypothesis, the α_{QP} are estimated by auxiliary ordinary least-squares regression using,

$$\alpha_{QP} = 1 + E[\frac{(x - E[x])^2 - x}{E[x]}]$$

with $E[x] = \mu =$ Poisson estimate for the expected value of the counts.

QP variance as a special case of NB variance and the manifestation of a non-decreasing relationship between θ and μ

It is instructive and useful to view QP variance as a special case of NB variance wherein α_{NB} exhibits dependence on μ through the following relationship

$$\theta = \frac{1}{\alpha_{NB}} = \frac{\mu}{\alpha_{QP} - 1} \tag{18}$$

Based on the relation above, we expect that if the values of α_{QP} for genes with μ within a given range of mean expression levels do not exhibit a dependence on their μ (genes with larger μ do not necessarily have larger α_{QP}), then we would observe a monotonic dependence between θ and μ ($\theta \propto \mu$) for the genes with means within that range of mean expression levels.

Using Eq. 18, we obtained estimates of θ for each gene using the corresponding values of α_{QP} and μ. We plotted the θ vs μ log-log plots for the genes using these values. For `Svensson 1` (left panel in Additional file 1: Fig. S1C), we observe a clear non-decreasing relationship between θ and μ, particularly for genes with low mean expression levels. In order to make the monotonic increase even more apparent, we plotted estimates of θ obtained by fixing α_{QP} to the value of α_{QP} for the gene with the highest point density estimate in the the θ vs μ log-log plot (the dashed red line). Furthermore, we plotted contour lines (white curves) based on the point densities as a visual aid to infer how the density of the points (genes) varies depending on μ. If we imagine a closed contour loop as an ellipse, the straight line through the center of the ellipse that joins the two points furthest from the center is called the major axis of the ellipse. We refer to analogous lines for the contours as their longer axes. Supposing no dependence of θ on μ, the longer axes of the contours would be parallel to the x—axis (zero slope). Instead, we observe that the dashed red line of the estimated θ with fixed value of α_{QP} lies along the same direction as the longer axes of the contours, particularly for genes located in the region with high point density (bright yellow region). For `PBMC 33k` (right panel in Additional file 1: Fig. S1C), once again, despite greater variability in θ due to the inherent biological variability, the monotonic increase in θ with μ particularly for genes with low mean expression levels is still very evident. Particularly for genes located in the region with high point density (bright yellow region), the dashed red line of the estimated θ with fixed α_{QP} closely follows the direction of the longer axes of the contours. The `NIH/3T3` data provides an insightful contrast (middle panel in Fig. 1C) - while the monotonic increase in θ with increase in μ is evident for genes with low mean expression levels, there is a clear discrepancy between the slope of the dashed red line (estimated θ with fixed value of α_{QP}) and the slope of the longer axes of the contours, particularly for genes in the region with the highest point density (bright yellow region). A closer examination reveals that for this data set the mean expression level (μ) of genes in the region of highest point density lie between 0.1 and 1, which is an order of magnitude higher than what we observe for `Svensson 1` and `PBMC 33k` (μ lies between 0.01 and 0.1 for genes in their respective regions of highest density). This difference actually stems from the differences in their respective sequencing depths - while the median total UMIs per cell in `Svensson 1` and `PBMC 33k` are 2309 and 1891 respectively, it is 15560 in `NIH/3T3`.

We must mention here that scTransform [20] relies on this non-decreasing relationship between gene abundance (μ) and the inverse over-dispersion parameter (θ) to perform regularization. In a recent paper [29], Choudhary and Satija argued that when modeling scRNA-seq data using a Gamma-Poisson distribution the inverse over-dispersion parameter (θ) does vary as a function of the gene abundance (μ), but that the true nature of this relationship can be masked for genes with low molecular counts. Their justification for such a relationship primarily rests on observations made for bulk RNAseq studies. However, for scRNA-seq counts data there appears to be a simpler explanation not linked to any underlying biological cause, namely the Poisson-like variance of counts for genes with low mean expression levels. This QP variance for low abundance genes holds not just for biological datasets but also for negative control datasets thus suggesting that there is no biological source for this non-decreasing relationship.

Hoes does the number of bins impact the feature selection process? And what about ribosomal genes?

The choice of number of bins plays an important role in the feature selection process. Fewer bins would lead to more genes per bin and this would result in an underestimation of the value of the quantile, especially for the bin that contains genes with the largest mean expression levels since their α_{QP} vary the most with μ. We show this in Additional file 1: Fig. S13 for the `PBMC 33k` data set. When we pick the number of bins to be 10 (top left panel in Additional file 1: Fig. S13), we observe a noticeable increase in the values of $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ with μ for genes with the highest mean expression levels. This will lead to a bias towards selection of high expression genes as HVGs. We can increase the number of bins to reduce this bias since this will ensure that genes with comparable mean expression levels are being grouped together (see bottom left panel for 1000 bins). However, if we keep on increasing the number of bins there will be very few genes per bin. Since we effectively assume that in every bin there is a stable gene we would end up concluding for a significant proportion of genes that these genes are stable even if they actually exhibit biological variability (note the decrease in values of $\alpha_{QP} - \alpha_{QP(Reference|Bin)}$ for genes with high mean expression levels in the bottom right panel; FTH1 for instance goes from having a value above 40 when the number of bins was set to 1000 to a value below 40 with the number of bins set to 10000). In practice, setting the number of bins to the default value of 1000 typically leads to 10-15 genes per bin with the gene with the smallest or the second smallest value within the bin providing the $\alpha_{QP(Reference|Bin)}$ value for that bin.

Hafemeister and Satija [20] noted earlier that the highly variable genes should not include housekeeping genes such as the ribosomal genes. This was based on the rationale that these genes while expressed at high levels are not expected to be variable across cell-types. However, it is increasingly being recognized that there can be tissue and development-stage specific variation in the transcription of the ribosomal genes [61]. Our feature selection method identifies genes as highly variable as long as they exhibit higher variability in their counts relative to other genes with comparable mean expression levels. Thus, we expect to observe variable numbers of ribosomal genes among the HVGs depending on the nature of the cells constituting the data set. Indeed, we observed that in the `Mouse Cortex (DroNC-seq)` data set only 14 out of 85 ribosomal genes were shortlisted among the top 3000 HVGs, while as many as 72 out of 85 ribosomal genes were present among the top 3000 HVGs for the NIH/3T3 mouse fibroblast cell line data. In the case of the latter, since the data set comprises a homogenous cell population we don't expect to see much variability overall, and therefore within this context where not many genes are expected to exhibit variability the shortlisting of ribosomal genes among the variable is not surprising since the notion of variability in our feature selection approach is relative and not absolute.

The standard estimate of cell-specific size factors assumes that the counts are Poisson distributed

There is an intimate link between the estimates of size factors given by Eq. (6) and the estimates for expected means ($\hat{\mu}_{gc}$) under the assumption that the counts are Poisson distributed. Assuming that each gene g contributes a proportion p_g of the total count N_c in cell c, the counts X_{gc} are modeled as,

$$X_{gc} \sim Poisson(\mu_{gc}) \tag{19}$$

where $\mu_{gc} = p_g N_c$. As shown by Townes et al. [27] and Lause et al. [21], the maximum likelihood estimates for p_g and N_c under the Poisson model are given by,

$$\hat{N}_c = \sum_g X_{gc} \tag{20}$$

$$\hat{p}_g = \frac{\sum_c X_{gc}}{\sum_c \hat{N}_c} \tag{21}$$

Based on these estimates, the maximum likelihood estimate for μ_{gc} is given by,

$$\hat{\mu}_{gc} = \frac{\sum_c X_{gc} . \sum_g X_{gc}}{\sum_c \sum_g X_{gc}} \tag{22}$$

If we divide both the numerator and denominator in Eq. (22) by the total number of cells (C), we get,

$$\hat{\mu}_{gc} = \frac{(\sum_c X_{gc}/C) . \sum_g X_{gc}}{(\sum_c \sum_g X_{gc})/C}$$

Since $\sum_c X_{gc}/C = \mu_g =$ mean of the observed counts of gene g,

$$\hat{\mu}_{gc} = \frac{\mu_g . \sum_g X_{gc}}{(\sum_c \sum_g X_{gc})/C}$$

Using Eq. (6), we can rewrite the above equation more simply as,

$$\hat{\mu}_{gc} = \mu_g SF_c \tag{23}$$

From this we can conclude that the simple estimates of the size factors given by Eq. (6) should be more appropriately viewed as estimates under the approximation that the counts are Poisson distributed.

Estimates for mean and variance under variance stabilization transformation

In order to compute z-scores for our data, we first need to apply a variance stabilization transformation to the observed counts (X_{gc}) to bring their distribution closer to the normal distribution. The variance stabilization transformation can be performed using monotonic non-linear functions, $g(X)$, such that the transformed counts are given by,

$$Y = g(X)$$

To compute the residuals, we need estimates for the means and variances of Y based on estimates for means and variances of X. We can arrive at approximations for both using a Taylor expansion around $X = \mu$,

$$Y \approx g(\mu) + g'(\mu)(X - \mu) + \frac{1}{2!}g''(\mu)(X - \mu)^2 + ...$$

where $g'(\mu)$ and $g''(\mu)$ are the first and second order derivatives of $g(X)$ evaluated at $X = \mu$. Considering the expansion up till the 1st order,

$$Y \approx g'(\mu)X + g(\mu) - g'(\mu)\mu$$

The expected value of Y can be approximated to,

$$E[Y] \approx g'(\mu)\mu + g(\mu) - g'(\mu)\mu = g(\mu) \tag{24}$$

In addition, since

$$Y - g(\mu) \approx g'(\mu)(X - \mu)$$

after squaring and taking expectation we get the following approximation for the variance of Y,

$$Var[Y] = Var[g(X)] \approx (g'(\mu))^2 Var[X] \tag{25}$$

For $g(X) = log(X + 1)$, the first order approximations of the mean and variance are then given by,

$$E[log(X + 1)] \approx log(\mu + 1) \tag{26}$$

$$Var[log(X + 1)] \approx \frac{1}{(\mu + 1)^2}\sigma^2 \tag{27}$$

Limitation of residuals without variance stabilization

We discuss the limitation of residuals computed for raw untransformed counts by considering the analytic Pearson residuals approach proposed by Lause et al. [21]. They proposed that the expected means ($\hat{\mu}_{gc}$) can be approximated with the estimates given by Eq. (22). Further, they argued that the over-dispersion coefficient ($\hat{\alpha}_g$) can be approximated with a fixed value of $\hat{\alpha}_g = 0.01$ corresponding to the typical over-dispersion observed in technical control datasets. Their rationale for these choices is that the null model should correspond to the measurement process so that the residuals provide estimates for deviations compared to expectations under the measurement model. While conceptually sound, this approach compromises variance stabilization, especially for genes that are robustly expressed in only a subset of cells while showing negligible expression in the rest of the cells (such genes would be considered as *markers* of the specific cell sub-populations in which they are expressed).

We can illustrate the nature of the problem with the help of a simplified example. Suppose we have a data set consisting of 10000 cells with a gene (gene A) that is only expressed in 100 cells with identical counts of 100 in each of those cells; the rest of the cells have 0 counts. For simplicity, if we assume that all the cells have the same sequencing depth ($SF_c = 1 \, \forall \, c$) then,

$$\hat{\mu}_{gc} = \mu_g$$

Based on this, Eq. (8) simplifies to,

$$r_{gc} = \frac{X_{gc} - \mu_g}{\sqrt{\mu_g + \hat{\alpha}_g \mu_g^2}}$$

Given the distribution of counts for gene A stated above, $\mu_A = 1$, and since we assume $\hat{\alpha}_g = 0.01$, the residuals of gene A for the cells with counts of 100 are approximately 98.509, while the residuals for cells with 0 counts are -0.995. The overall variance of the residuals for gene A is approximately 98.03, thus exhibiting significant deviation from the null expectation of 1. The Pearson residuals method proposed by Hafemeister et al. [20] that allows for per gene estimates for $\hat{\alpha}_g$ addresses this issue only to a limited extent.

Variance stabilization transformations implemented in Piccolo

Apart from the *log* transformation as the variance stabilization transformation discussed in the main text and implemented as the default option in `Piccolo`, we also offer the option to apply two other transformations which are described below:

Sqrt

For the *sqrt* transformation, $g(X) = \sqrt{X}$. The first order approximations of the mean and variance under this transformation are,

$$E[\sqrt{X}] \approx \sqrt{\mu}$$

$$Var[\sqrt{X}] \approx \frac{\sigma^2}{1/(4 * \mu)}$$

After accounting for sampling depth differences ($\hat{\mu}_{gc} = SF_c \tilde{\mu}_g$ and $\hat{\sigma}_{gc}^2 = SF_c^2 \tilde{\sigma}_g^2$) we have,

$$E[\sqrt{X_{gc}}] \approx \sqrt{\hat{\mu}_{gc}}$$

$$Var[\sqrt{X_{gc}}] \approx \frac{\hat{\sigma}_{gc}^2}{1/(4 * \hat{\mu}_{gc})}$$

Based on these first order approximations for means and variances under the $\sqrt{X}$ transformation, the residuals are given by,

$$Z'_{gc} = \frac{\sqrt{X_{gc}} - \sqrt{\hat{\mu}_{gc}}}{\hat{\sigma}_{gc}/(1/(2 * \sqrt{\hat{\mu}_{gc}}))}$$

Box-Cox Power Law Transform

The Box-Cox transform [62] belongs to the family of power law transformations. Power law transformations are applicable only for positive variables and are indexed by a parameter λ, such that for an arbitrary observation x, the transformed value is given by x^{λ}. The Box-Cox transformation is a modified power transformation defined as,

$$g(X = x) = x^{(\lambda)} = \frac{x^{\lambda} - 1}{\lambda}, \lambda \neq 0$$

which is a continuous function in λ for $x > 0$ ($g(X) = x^{(\lambda)} = log(x)$, when $\lambda = 0$). Box-Cox proposed that based on the observations x_1, x_2,..., x_n, the appropriate choice of λ corresponds to the value that maximizes,

$$l(\lambda) = -\frac{n}{2} \log[\frac{1}{n} \sum_{j=1}^{n} (x_j^{(\lambda)} - \overline{x^{(\lambda)}})^2] + (\lambda - 1) \sum_{j=1}^{n} \log x_j$$

where,

$$\overline{x^{(\lambda)}} = \frac{1}{n} \sum_{j=1}^{n} x_j^{(\lambda)} = \frac{1}{n} \sum_{j=1}^{n} (\frac{x_j^{\lambda} - 1}{\lambda})$$

For the Box-Cox transformation, the first-order approximations of the mean and variance are,

$$E[g(X)] \approx \frac{(\mu + 1)^{\lambda} - 1}{\lambda}$$

$$Var[g(X)] \approx \sigma^2 * (\mu + 1)^{\lambda - 1}$$

where pseudo-counts of 1 have been added to ensure positive (non-zero) values. After accounting for sampling depth differences ($\hat{\mu}_{gc} = SF_c \tilde{\mu}_g$ and $\hat{\sigma}^2_{gc} = SF_c^2 \tilde{\sigma}_g^2$) we have,

$$\begin{aligned} E[g(X_{gc})] &\approx \frac{(\hat{\mu}_{gc} + 1)^{\lambda} - 1}{\lambda} \\ Var[g(X_{gc})] &\approx \hat{\sigma}^2_{gc} * (\hat{\mu}_{gc} + 1)^{\lambda - 1} \end{aligned}$$

Based on these first-order approximations for means and variances under the Box-Cox transformation, the residuals are given by,

$$Z'_{gc} = \frac{\frac{(X_{gc}+1)^{\lambda}-1}{\lambda} - \frac{(\hat{\mu}_{gc}+1)^{\lambda}-1}{\lambda}}{\hat{\sigma}_{gc} * \sqrt{(\hat{\mu}_{gc} + 1)^{\lambda - 1}}}$$

LogSF

We also provide the popular `logSF` normalization as an option in `Piccolo`. The normalized counts under this transformation are given by,

$$X_{gc}^{'} = \log(\frac{X_{gc}}{SF_c} + 1)$$

where estimates of SF_c are obtained using stable genes (see eq. (7))

We compared these variance stabilization transformations within Piccolo by applying them to the 100 subsets created using the `Zheng Mix 8eq` data set. The results are shown in Additional file 1: Fig. S23. The Box-Cox transform performs the best overall. However, the trade-off is that it is also the most computationally intensive. Nevertheless, we would still recommend users to consider applying it for small or medium sized datasets ($10^3 - 10^5$ cells).

Differential expression analysis in Piccolo

After applying our normalization to the observed counts, the distribution of the residuals are brought closer to the normal distribution due to the variance stabilization. This makes it possible to employ the two-sample Student's t-test with the null hypothesis that the means of the two samples are the same. Ideally, the test is applicable only if the variances of the two samples can be assumed to be equal. The variance stabilization during normalization ensures that under most circumstances this assumption holds true.

Typically, the preferred test for single-cell differential expression analyses is the Wilcoxon rank-sum test which is a non-parametric alternative to the two-sample t-test. A key difference between the two is that while the t-test actually tests for location shifts (differences in means) between the samples, the Wilcoxon rank-sum test can be sensitive to shifts in distribution other than a pure location shift.

Obtaining corrected counts from the *z*-scores

While the clustering and the differential expression analyses are performed using the z-scores obtained from `Piccolo`, in some other applications and for the purpose of applying other tools we can also obtain estimates of both the *log*-transformed values as well as the corrected counts. We illustrate how this is done for `Piccolo` normalization (*log*-based variance stabilization).

Recall that,

$$\tilde{\mu}_g = \frac{1}{C}\sum_c \frac{X_{gc}}{SF_c}$$

$$\tilde{\sigma}_g^2 = \frac{1}{C-1}\sum_c (\frac{X_{gc}}{SF_c} - \tilde{\mu}_g)^2$$

are the respective estimates for the gene mean and variance after accounting for sampling depths. Based on these, the estimates for the *log*-transformed values can be obtained using Eq. (16) as,

$$\log(\tilde{X}_{gc} + 1) = Z'_{gc} * \frac{\tilde{\sigma}_g}{\tilde{\mu}_g + 1} + \log(\tilde{\mu}_g + 1)$$

Using these *log*-transformed values, we can then obtain estimates for the corrected counts,

$$\tilde{X}_{gc} = round(\exp(Z'_{gc} * \frac{\tilde{\sigma}_g}{\tilde{\mu}_g + 1} + \log(\tilde{\mu}_g + 1)) - 1)$$

where the rounding ensures that the corrected counts have integer values.

Batch effect correction

Batch effects are technical confounders that typically stem from a wide array of non-biological sources such as differences in the reagents, the individuals handling and processing the samples, as well as the time of experiment. If not accounted for, batch effect can be misinterpreted as a biological signal and lead to erroneous conclusions. The z-score based normalization of `Piccolo` allows for a very simplistic approach to perform batch effect correction based on the fundamental assumption that all the biological conditions are processed in all the batches (Note: It is tempting to apply batch correction even when this assumption is not met, however the conclusions drawn from such analyses will be unreliable).

The basic idea of our batch effect correction method is to identify HVGs independently for each batch, and then seek the HVGs that match between the batches. This ensures that we only retain those genes for downstream analysis that are variable in every batch, and eliminate genes that exhibit variability of counts only within specific batches since the latter likely reflect batch-specific effects. In addition, we identify stable genes across all batches as well as stable genes individually for each batch. We retain only those stable genes in each batch that are also stable across the batches. This ensures that the counts are normalized using size factors estimated with stable genes in each batch that we know are less likely to exhibit batch-specific variation. We then perform the z-score normalization independently for each batch. PCA is applied on the composite z-scores matrix obtained by combining the z-scores matrices from all the batches.

We provide an illustration of the application of our simplistic batch effect correction on a mouse intestinal epithelium data set (`Haber 2017`) that consists of cells obtained in 10 batches [59] (see Additional file 1: Fig. S24). Further, we used Splat [39] to simulate a data set using `NIH/3T3` that contained 5 groups of cells, with half the cells in one batch, and the other half simulated to belong to another batch (see Additional file 1: Fig. S25). With the help of these examples, we are able to show that the simplistic batch correction approach works well when the assumption that all the conditions are present in every batch is reasonably satisfied.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-024-05872-w.

Supplementary Material 1. Supplementary Tables and Figures

Acknowledgements
We thank the members of the Khiabanian and Herranz labs for their encouragement and insightful feedback. Special thanks are due to Mona Arabzadeh and Vaidhyanathan Mahaganapathy for carefully reading the manuscript and suggesting improvements.

Author contributions
AS conceived and performed all the analyses. AS implemented the Piccolo R package. AS and HK prepared the manuscript.

Funding
This work was supported by grants from the National Institutes of Health (R01CA233662 and P30CA072720), the V Foundation (T2019-01), and the New Jersey Commission on Cancer Research (COCR23PRG006 & COCR24PDF015).

Availability of data and materials
No original dataset was created as a part of this study. All the publicly available datasets discussed in this manuscript are listed and cited in the table in the Datasets section along with links to the primary sources from where they were obtained. In order to make it easier to reproduce the analyses in this manuscript, we have also provided access to all the datasets together with all the scripts at one location: https://github.com/Amartya101/PiccoloPaperData.
The development version of our R package, `Piccolo`, is available through GitHub at https://github.com/Amartya101/Piccolo, and is licensed under the GNU GPLv3 license. In order to facilitate use with existing tools, the feature selection and normalization methods implemented in `Piccolo` can be used with Seurat (version 5) [18, 60], the instructions for which are provided at https://github.com/Amartya101/Piccolo-With-Seurat

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
No competing interests to declare.

Received: 4 July 2024 Accepted: 16 July 2024
Published online: 30 July 2024

References

1. Grün D, Lyubimova A, Kester L, Wiebrands K, Basak O, Sasaki N, Clevers H, van Oudenaarden A. Single-cell messenger RNA sequencing reveals rare intestinal cell types. Nature. 2015;525(7568):251–5.
2. Villani AC, Satija R, Reynolds G, Sarkizova S, Shekhar K, Fletcher J, Griesbeck M, Butler A, Zheng S, Lazo S, Jardine L, Dixon D, Stephenson E, Nilsson E, Grundberg I, McDonald D, Filby A, Li W, De Jager PL, Rozenblatt-Rosen O, Lane AA, Haniffa M, Regev A, Hacohen N. Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. Science. 2017. https://doi.org/10.1126/science.aah4573.
3. Shalek AK, Satija R, Adiconis X, Gertner RS, Gaublomme JT, Raychowdhury R, Schwartz S, Yosef N, Malboeuf C, Diana L, Trombetta JJ, Gennert D, Gnirke A, Goren A, Hacohen N, Levin JZ, Park H, Regev A. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. Nature. 2013;498(7453):236–40.
4. Shalek AK, Satija R, Shuga J, Trombetta JJ, Gennert D, Diana L, Chen P, Gertner RS, Gaublomme JT, Yosef N, Schwartz S, Fowler B, Weaver S, Wang J, Wang X, Ding R, Raychowdhury R, Friedman N, Hacohen N, Park H, May AP, Regev A. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. Nature. 2014;510(7505):363–9.
5. Torre E, Dueck H, Shaffer S, Gospocic J, Gupte R, Bonasio R, Kim J, Murray J, Raj A. Rare cell detection by single-cell RNA sequencing as guided by single-molecule RNA FISH. Cell Syst. 2018;6(2):171-179.e5.
6. Treutlein B, Brownfield DG, Wu AR, Neff NF, Mantalas GL, Hernan Espinoza F, Desai TJ, Krasnow MA, Quake SR. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. Nature. 2014;509(7500):371–5.
7. Bach K, Pensa S, Grzelak M, Hadfield J, Adams DJ, Marioni JC, Khaled WT. Differentiation dynamics of mammary epithelial cells revealed by single-cell RNA sequencing. Nat Commun. 2017;8(1):2128.
8. Ding J, Adiconis X, Simmons SK, Kowalczyk MS, Hession CC, Marjanovic ND, Hughes TK, Wadsworth MH, Burks T, Nguyen LT, Kwon JYH, Barak B, Ge W, Kedaigle AJ, Carroll S, Li S, Hacohen N, Rozenblatt-Rosen O, Shalek AK, Villani A-C, Regev A, Levin JZ. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. Nat Biotechnol. 2020;38(6):737–46.
9. Andrews TS, Kiselev VY, McCarthy D, Hemberg M. Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. Nat Protoc. 2021;16(1):1–9.
10. Svensson V. Droplet scRNA-seq is not zero-inflated. Nat Biotechnol. 2020;38(2):147–50.
11. Sarkar A, Stephens M. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. Nat Genet. 2021;53(6):770–7.
12. Lun ATL, McCarthy DJ, Marioni JC. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with bioconductor. F1000Res. 2016;5:2122.

13. Amezquita RA, Lun ATL, Becht E, Carey VJ, Carpp LN, Geistlinger L, Marini F, Rue-Albrecht K, Risso D, Soneson C, Waldron L, Pagès H, Smith ML, Huber W, Morgan M, Gottardo R, Hicks SC. Orchestrating single-cell analysis with bioconductor. Nat Methods. 2020;17(2):137–45.
14. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol. 2019;15(6): e8746.
15. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 2015;33:495–502.
16. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat Biotechnol. 2018;36:411–20.
17. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, Yuhan H, Marlon S, Peter S, Rahul S. Comprehensive integration of single-cell data. Cell. 2019;177:1888–902.
18. ...Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, Lee MJ, Wilk AJ, Darby C, Zager M, Hoffman P, Stoeckius M, Papalexi E, Mimitou EP, Jain J, Srivastava A, Stuart T, Fleming LB, Yeung B, Rogers AJ, McElrath JM, Blish CA, Gottardo R, Smibert P, Satija R. Integrated analysis of multimodal single-cell data. Cell. 2021. https://doi.org/10.1016/j.cell.2021.04.048.
19. Alexander Wolf F, Angerer P, Theis FJ. Scanpy: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19(1):15.
20. Hafemeister C, Satija R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. Genome Biol. 2019;20(1):296.
21. Lause J, Berens P, Kobak D. Analytic pearson residuals for normalization of single-cell RNA-seq UMI data. Genome Biol. 2021;22(1):258.
22. Ahlmann-Eltze C, Huber W. Comparison of transformations for single-cell RNA-seq data. Nat Methods. 2023. https://doi.org/10.1038/s41592-023-01814-1.
23. Colin Cameron A, Trivedi PK. Regression-based tests for overdispersion in the Poisson model. J Econom. 1990;46(3):347–64.
24. Colin CA, Trivedi PK. Regression Analysis of Count Data. Econometric Society Monographs. Cambridge University Press, 2 edition, 2013.
25. Angelidis I, Simon LM, Fernandez IE, Strunz M, Mayr CH, Greiffo FR, Tsitsiridis G, Ansari M, Graf E, Strom T-M, Nagendran M, Desai T, Eickelberg O, Mann M, Theis FJ, Schiller HB. An atlas of the aging lung mapped by single cell transcriptomics and deep tissue proteomics. Nat Commun. 2019;10(1):963.
26. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with deseq2. Genome Biol. 2014;15(12):550.
27. William Townes F, Hicks SC, Aryee MJ, Irizarry RA. Feature selection and dimension reduction for single-cell RNA-seq based on a multinomial model. Genome Biol. 2019;20(1):295.
28. Kharchenko PV. The triumphs and limitations of computational methods for scRNA-seq. Nat Methods. 2021;18(7):723–32.
29. Choudhary S, Satija R. Comparison and evaluation of statistical error models for scRNA-seq. Genome Biol. 2022;23(1):27.
30. Dorfman RA. A note on the δ-method for finding variance formulae. Biom Bull. 1938;1:129–37.
31. Hotelling H. Relations between two sets of variates. Biometrika. 1936;28(3–4):321–77.
32. Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, Peshkin L, Weitz DA, Kirschner MW. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell. 2015;161(5):1187–201.
33. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep. 2019;9(1):5233.
34. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp. 2008;2008(10):P10008.
35. Duò A, Robinson MD, Soneson C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. F1000Res. 2018;7:1141.
36. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, Ziraldo SB, Wheeler TD, McDermott GP, Zhu J, Gregory MT, Shuga J, Montesclaros L, Underwood JG, Masquelier DA, Nishimura SY, Schnall-Levin M, Wyatt PW, Hindson CM, Bharadwaj R, Wong A, Ness KD, Beppu LW, Joachim Deeg H, McFarland C, Loeb KR, Valente WJ, Ericson NG, Stevens EA, Radich JP, Mikkelsen TS, Hindson BJ, Bielas JH. Massively parallel digital transcriptional profiling of single cells. Nat Commun. 2017;8(1):14049.
37. McInnes L, Healy J, James M. Umap:uniform manifold approximation and projection for dimension reduction; 2018.
38. Baruzzo G, Patuzzi I, Di Camillo B. SPARSim single cell: a count data simulator for scRNA-seq data. Bioinformatics. 2019;36(5):1468–75.
39. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. Genome Biol. 2017;18(1):174.
40. Qin F, Luo X, Xiao F, Cai G. SCRIP: an accurate simulator for single-cell RNA sequencing data. Bioinformatics. 2021;38(5):1304–11.
41. Cao Y, Yang P, Yang JYH. A benchmark study of simulation methods for single-cell RNA sequencing data. Nat Commun. 2021;12(1):6911.
42. Hahsler M, Piekenbrock M. dbscan: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Related Algorithms, 2022. R package version 1.1-11.
43. Hahsler M, Piekenbrock M, Doran D. dbscan: fast density-based clustering with R. J Stat Softw. 2019;91(1):1–30.
44. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2022.
45. Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K. cluster: Cluster Analysis Basics and Extensions, 2022. R package version 2.1.4 — For new features, see the 'Changelog' file (in the package source).
46. Dowle M, Srinivasan A. data.table: Extension of 'data.frame', 2021. R package version 1.14.2.
47. Wickham H. ggplot2: elegant graphics for data analysis. New York: Springer-Verlag; 2016.

48. Csardi G, Nepusz T. The igraph software package for complex network research. InterJournal, Complex Syst.2006;1695.
49. Bates D, Maechler M, Jagan M. Matrix: Sparse and Dense Matrix Classes and Methods, 2022. R package version 1.5-1.
50. Koncevicius K. matrixTests: Fast Statistical Hypothesis Tests on Rows and Columns of Matrices, 2021. R package version 0.1.9.1.
51. Qiu Y, Mei J. RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems, 2022. R package version 0.16-1.
52. van der Maaten LJP, Hinton GE. Visualizing high-dimensional data using t-sne. J Mach Learn Res. 2008;9:2579–605.
53. van der Maaten LJP. Accelerating t-sne using tree-based algorithms. J Mach Learn Res. 2014;15:3221–45.
54. Krijthe JH. Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation, 2015. R package version 0.16.
55. Konopka T. umap: Uniform Manifold Approximation and Projection, 2022. R package version 0.2.9.0.
56. Garnier S, Ross N, Rudis R, Camargo AP, Sciaini M, Scherer C. viridis - Colorblind-Friendly Color Maps for R, 2021. R package version 0.6.2.
57. Svensson V, Natarajan KN, Ly L-H, Miragaia RJ, Labalette C, Macaulay IC, Cvejic A, Teichmann SA. Power analysis of single-cell RNA-sequencing experiments. Nat Methods. 2017;14(4):381–7.
58. Angelo D, Charlotte S. DuoClustering2018: Data, Clustering Results and Visualization Functions From Duò et al (2018), 2022. R package version 1.14.0.
59. Haber AL, Biton M, Rogel N, Herbst RH, Shekhar K, Smillie C, Burgin G, Delorey TM, Howitt MR, Katz Y, Tirosh I, Beyaz S, Dionne D, Zhang M, Raychowdhury R, Garrett WS, Rozenblatt-Rosen O, Shi HN, Yilmaz O, Xavier RJ, Regev A. A single-cell survey of the small intestinal epithelium. Nature. 2017;551(7680):333–9.
60. Hao Y, Stuart T, Kowalski MH, Choudhary S, Hoffman P, Hartman A, Srivastava A, Molla G, Madad S, Fernandez-Granda C, Satija R. Dictionary learning for integrative, multimodal and scalable single-cell analysis. Nat Biotechnol. 2023. https://doi.org/10.1038/s41587-023-01767-y.
61. Panda A, Yadav A, Yeerna H, Singh A, Biehl M, Lux M, Schulz A, Klecha T, Doniach S, Khiabanian H, Ganesan S, Tamayo P, Bhanot G. Tissue- and development-stage-specific mRNA and heterogeneous CNV signatures of human ribosomal proteins in normal and cancer samples. Nucleic Acids Res. 2020;48(13):7079–98.
62. Box GEP, Cox DR. An analysis of transformations. J R Stat Soc Ser B (Methodol). 1964;26(2):211–52.

Publisher's Note